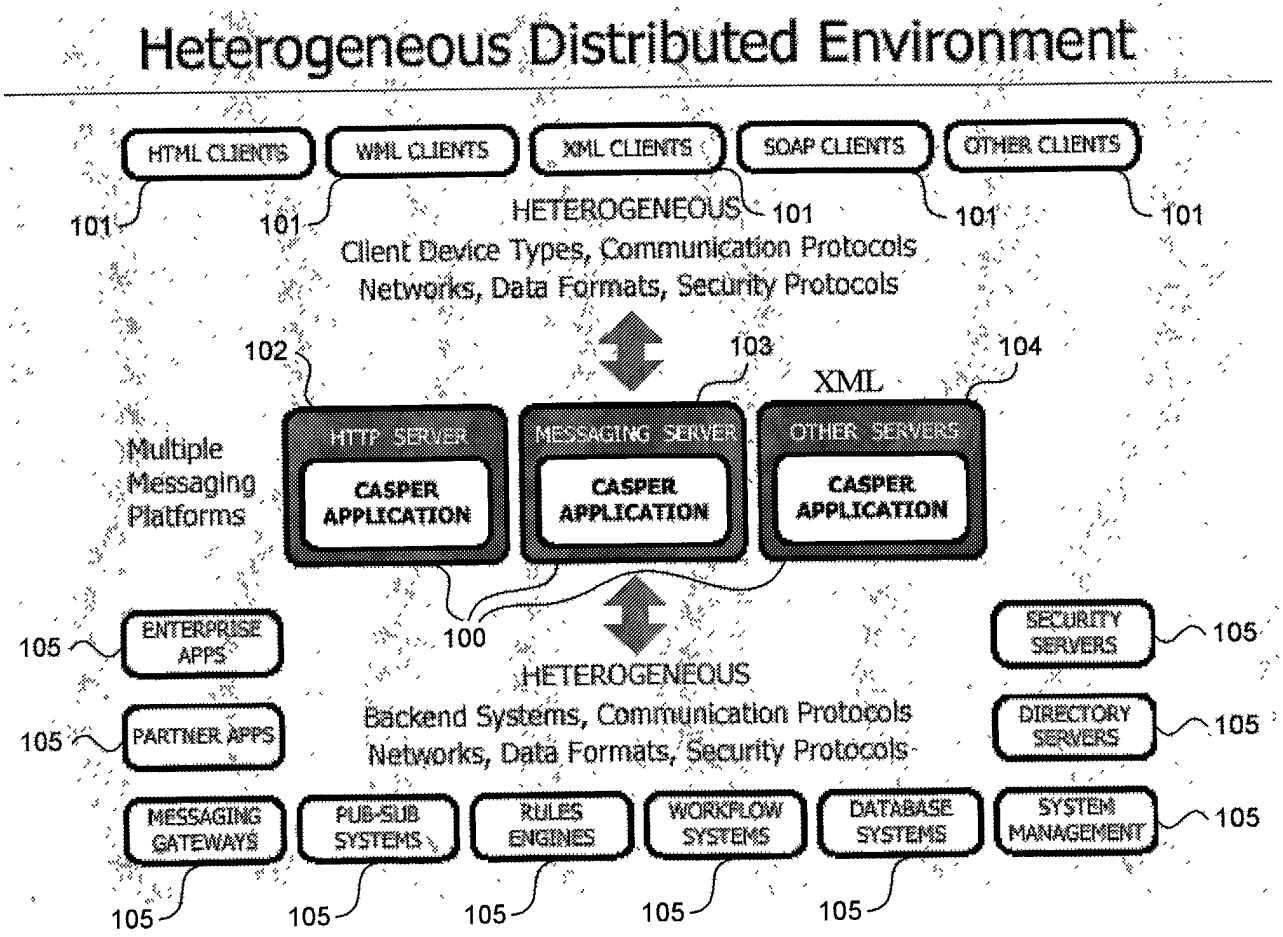

Heterogeneous Distributed Environment
HTML CLIENTS
WML CLIENTS
XML CLIENTS
SOAP CLIENTS
OTHER CLIENTS
101
101
101
101
101
HETEROGENEOUS
Client Device Types, Communication Protocols
Networks, Data Formats, Security Protocols
102
103
104
XML
Multiple
Messaging
Platforms
HTTP SERVER
CASPER APPLICATION
MESSAGING SERVER
CASPER APPLICATION
OTHER SERVERS
CASPER APPLICATION
100
ENTERPRISE APPS
105
PARTNER APPS
105
HETEROGENEOUS
Backend Systems, Communication Protocols
Networks, Data Formats, Security Protocols
SECURITY SERVERS
105
DIRECTORY SERVERS
105
MESSAGING GATEWAYS
105
PUB-SUB SYSTEMS
105
RULES ENGINES
105
WORKFLOW SYSTEMS
105
DATABASE SYSTEMS
105
SYSTEM MANAGEMENT
105

Fig. 1

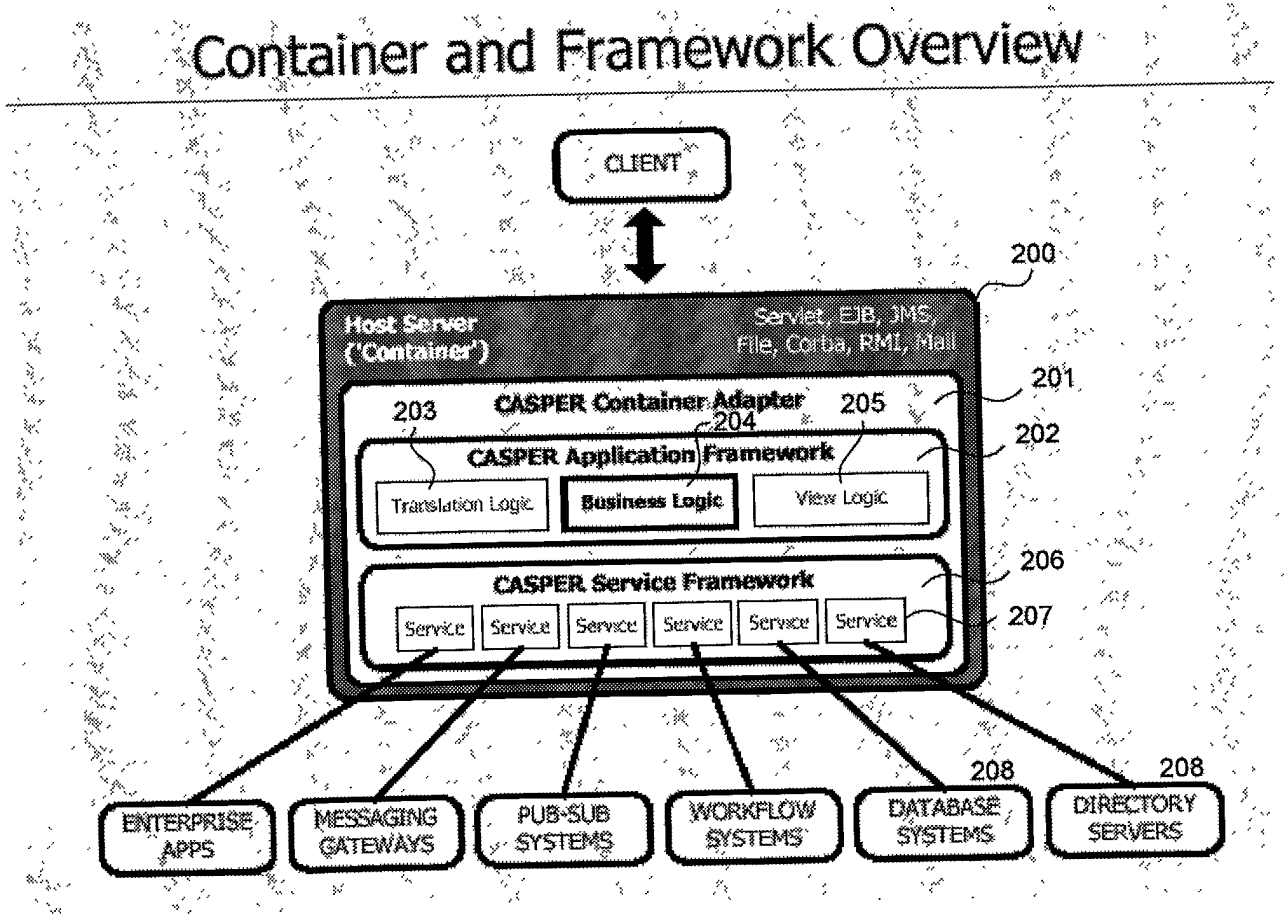

Container and Framework Overview
CLIENT
200
Host Server
('Container')
Servlet, EJB, JMS,
File, Corba, RMI, Mail
201
203
CASPER Container Adapter
205
204
202
CASPER Application Framework
Translation Logic
Business Logic
View Logic
206
CASPER Service Framework
Service
Service
Service
Service
Service
Service
207
208
208
ENTERPRISE APPS
MESSAGING GATEWAYS
PUB-SUB SYSTEMS
WORKFLOW SYSTEMS
DATABASE SYSTEMS
DIRECTORY SERVERS

Fig. 2

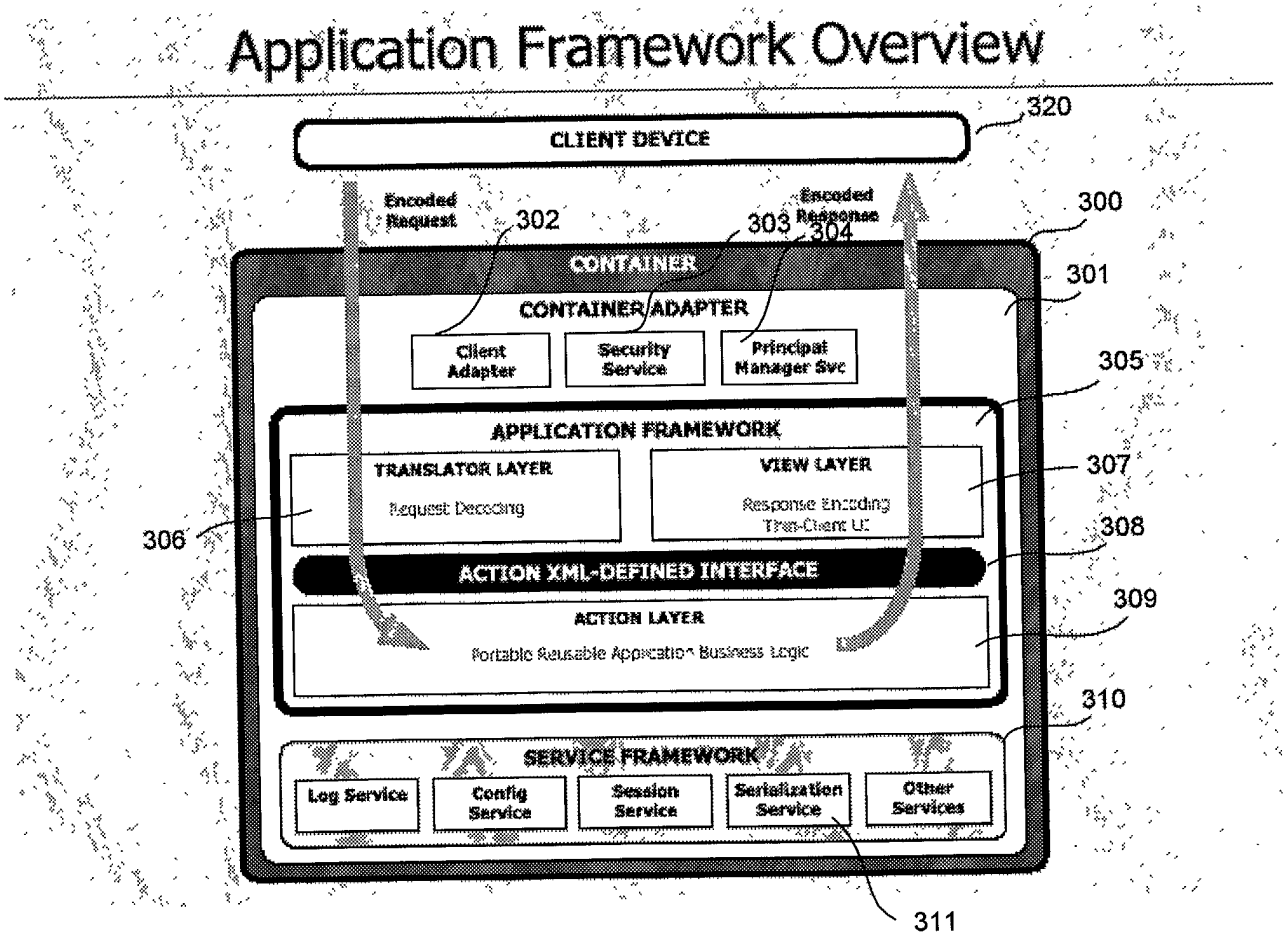

Application Framework Overview
CLIENT DEVICE
320
Encoded Request
302
303
304
Encoded Response
300
CONTAINER
301
CONTAINER/ADAPTER
Client Adapter
Security Service
Principal Manager Svc
305
APPLICATION FRAMEWORK
TRANSLATOR LAYER
Request Decoding
VIEW LAYER
Response Encoding
Thin-Client UI
307
306
308
ACTION XML-DEFINED INTERFACE
309
ACTION LAYER
Portable Reusable Application Business Logic
310
SERVICE FRAMEWORK
Log Service
Config Service
Session Service
Serialization Service
Other Services
311

Fig. 3

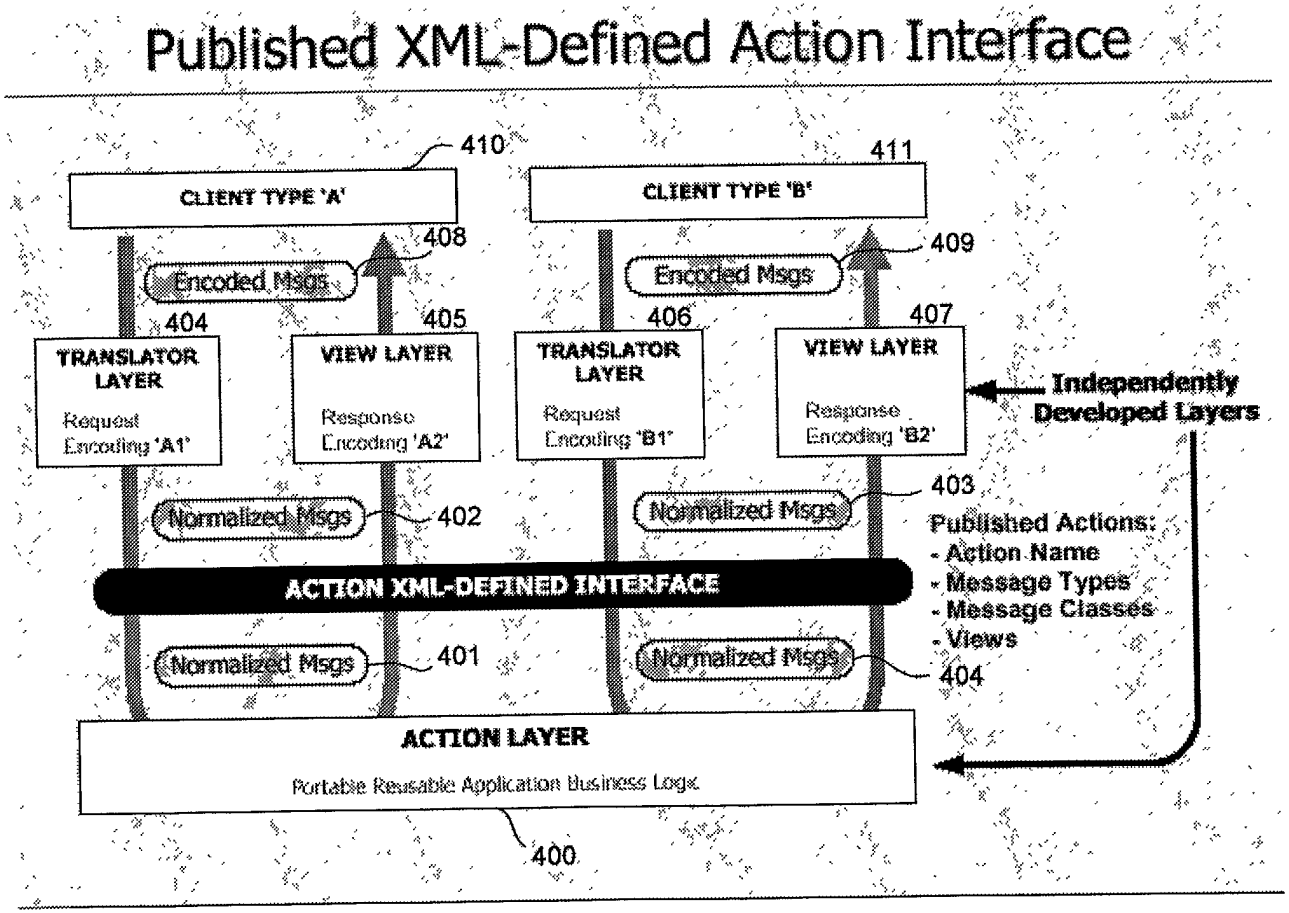

Published XML-Defined Action Interface
410
411
CLIENT TYPE 'A'
CLIENT TYPE 'B'
408
409
Encoded Msgs
Encoded Msgs
404
405
406
407
TRANSLATOR LAYER
Request Encoding 'A1'
VIEW LAYER
Response Encoding 'A2'
TRANSLATOR LAYER
Request Encoding 'B1'
VIEW LAYER
Response Encoding 'B2'
Independently Developed Layers
403
Normalized Msgs
402
Normalized Msgs
Published Actions:
- Action Name
- Message Types
- Message Classes
- Views
ACTION XML-DEFINED INTERFACE
Normalized Msgs
401
Normalized Msgs
404
ACTION LAYER
Portable Reusable Application Business Logic
400

Fig. 4

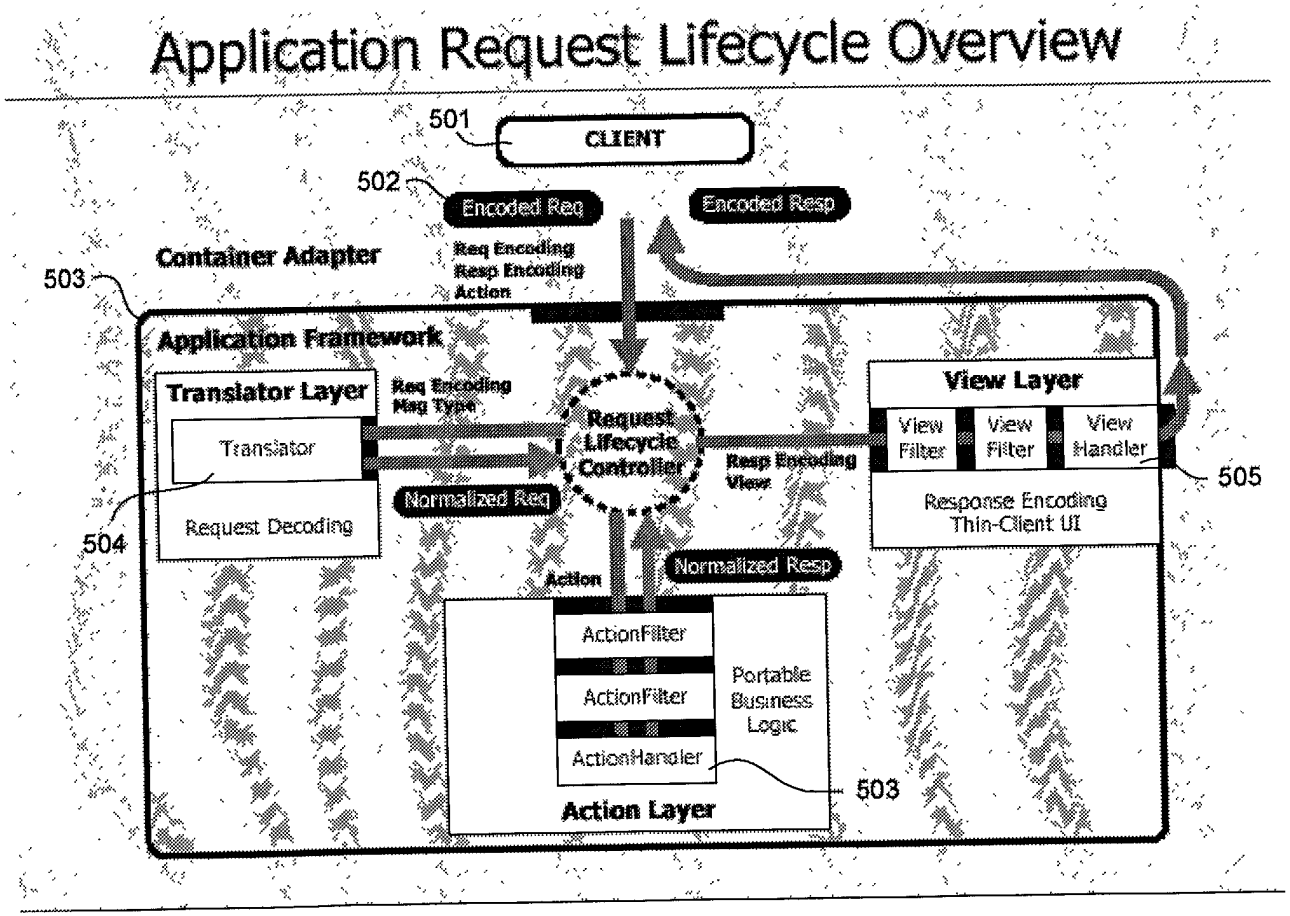

Application Request Lifecycle Overview
501
CLIENT
502
Encoded Req
Encoded Resp
Container Adapter
Req Encoding
Resp Encoding
Action
503
Application Framework
Translator Layer
Req Encoding
Msg Type
Translator
Request Decoding
504
Request Lifecycle Controller
Normalized Req
Resp Encoding
View
View Layer
View Filter
View Filter
View Handler
505
Response Encoding
Thin-Client UI
Action
Normalized Resp
ActionFilter
ActionFilter
ActionHandler
Portable Business Logic
Action Layer
503

Fig. 5

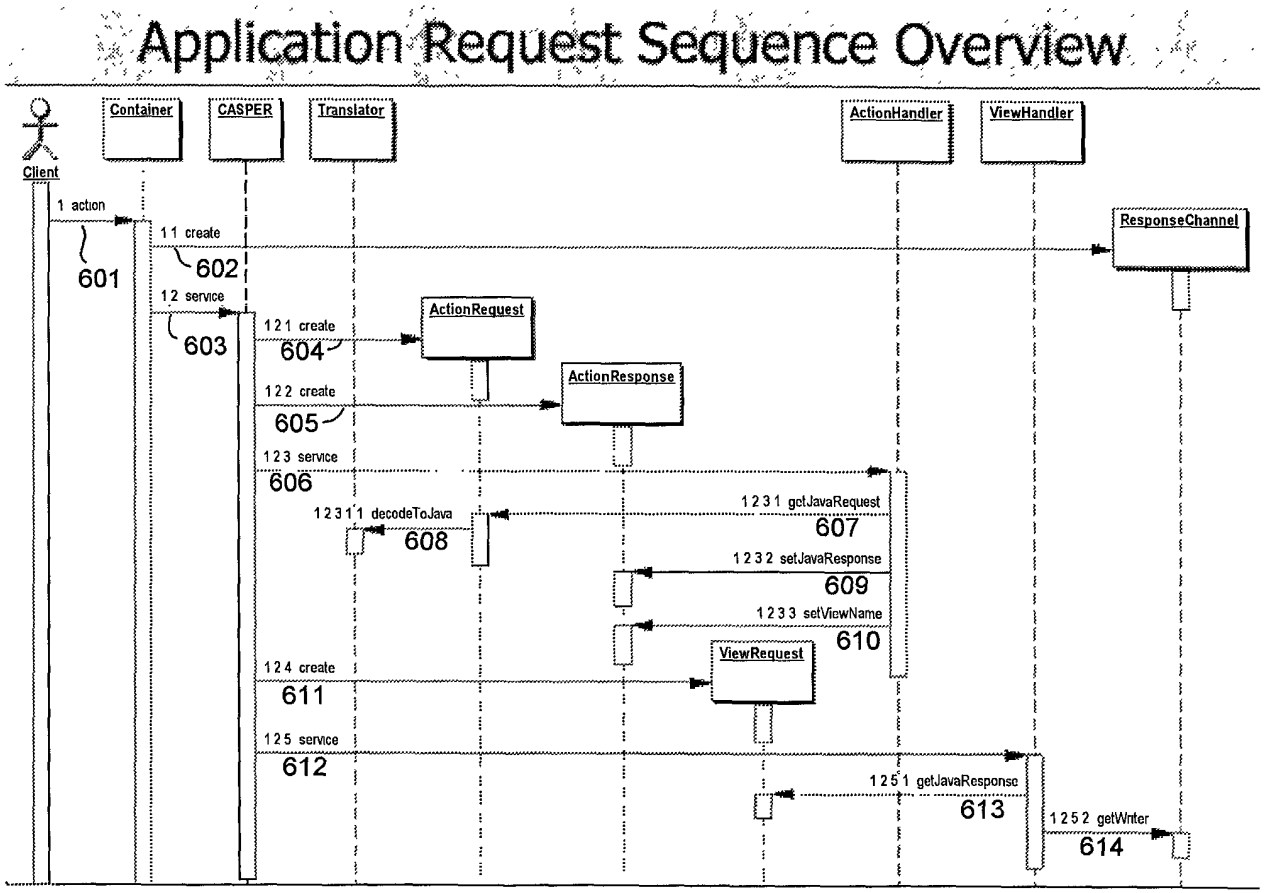

Application Request Sequence Overview
Client
Container
CASPER
Translator
ActionHandler
ViewHandler
ResponseChannel
ActionRequest
ActionResponse
ViewRequest
1 action
601
11 create
602
12 service
603
121 create
604
122 create
605
123 service
606
1231 getJavaRequest
607
12311 decodeToJava
608
1232 setJavaResponse
609
1233 setViewName
610
124 create
611
125 service
612
1251 getJavaResponse
613
1252 getWriter
614

Fig. 6

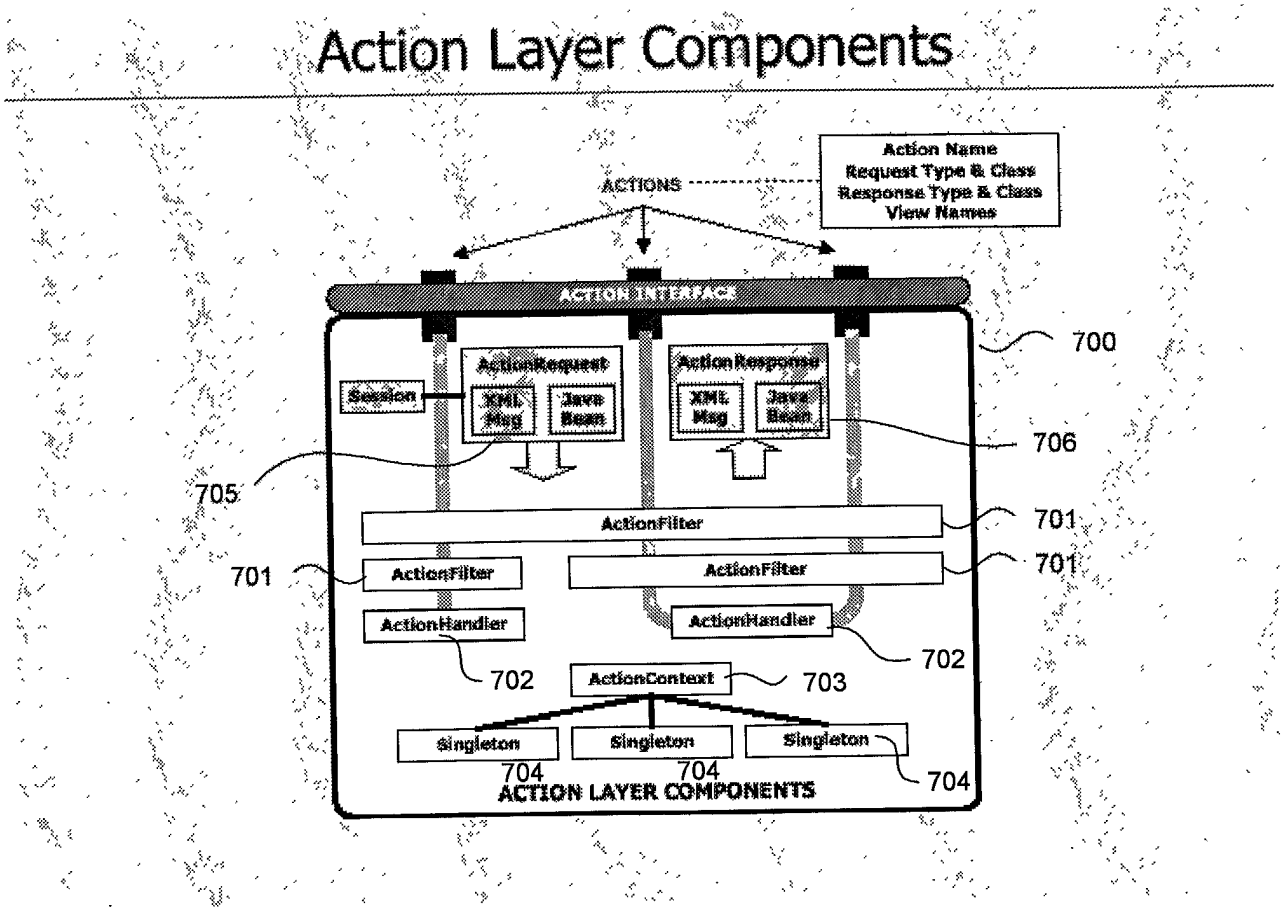

Action Layer Components
ACTIONS
Action Name
Request Type & Class
Response Type & Class
View Names
ACTION INTERFACE
700
ActionRequest
XML Msg
Java Bean
Session
ActionResponse
XML Msg
Java Bean
706
705
ActionFilter
701
ActionFilter
ActionFilter
701
701
ActionHandler
ActionHandler
702
702
ActionContext
703
Singleton
Singleton
Singleton
704
704
704
ACTION LAYER COMPONENTS

Fig. 7

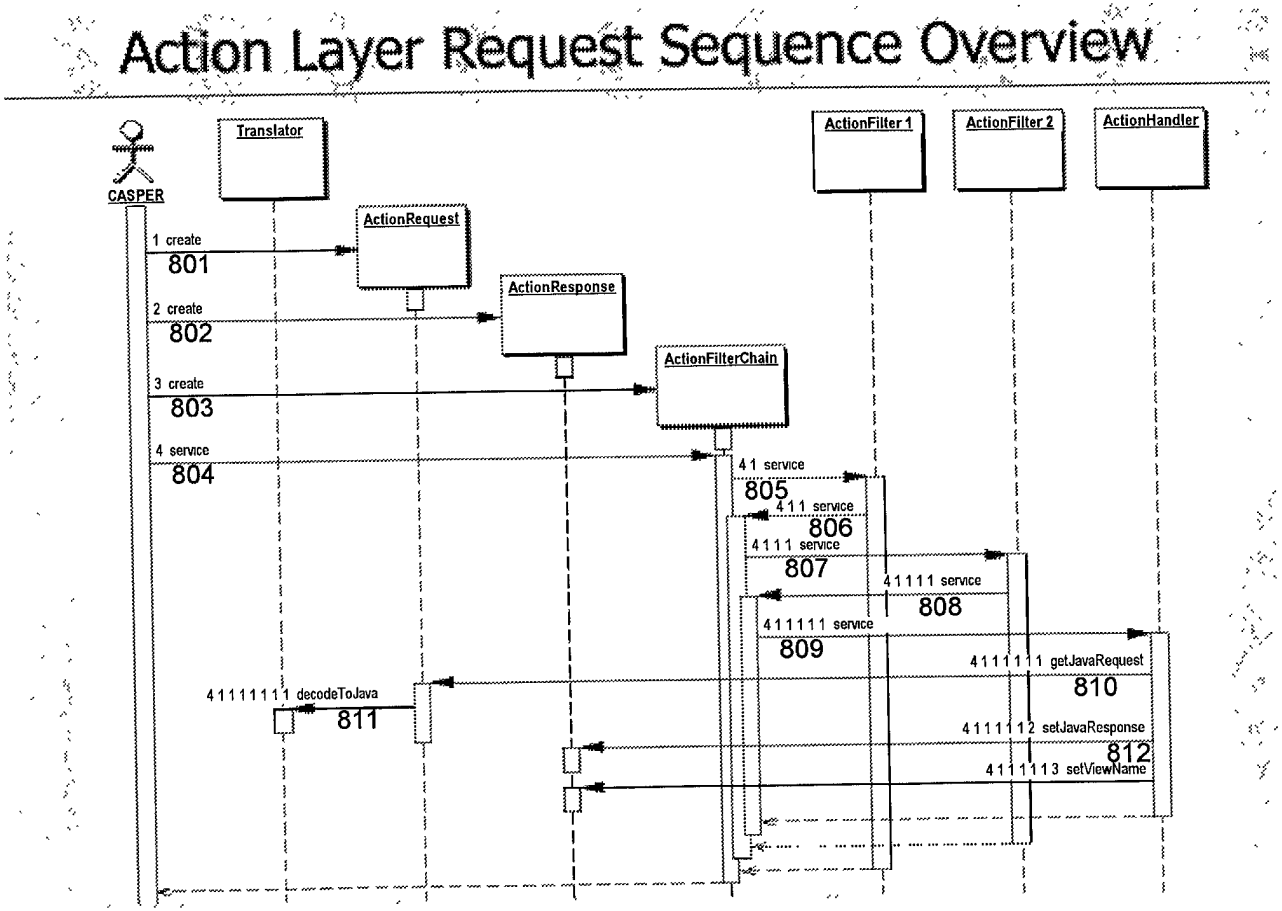

Action Layer Request Sequence Overview
CASPER
Translator
ActionRequest
ActionResponse
ActionFilterChain
ActionFilter 1
ActionFilter 2
ActionHandler
1 create
801
2 create
802
3 create
803
4 service
804
41 service
805
411 service
806
4111 service
807
41111 service
808
411111 service
809
4111111 getJavaRequest
810
41111112 setJavaResponse
812
4111113 setViewName
41111111 decodeToJava
811

Fig. 8

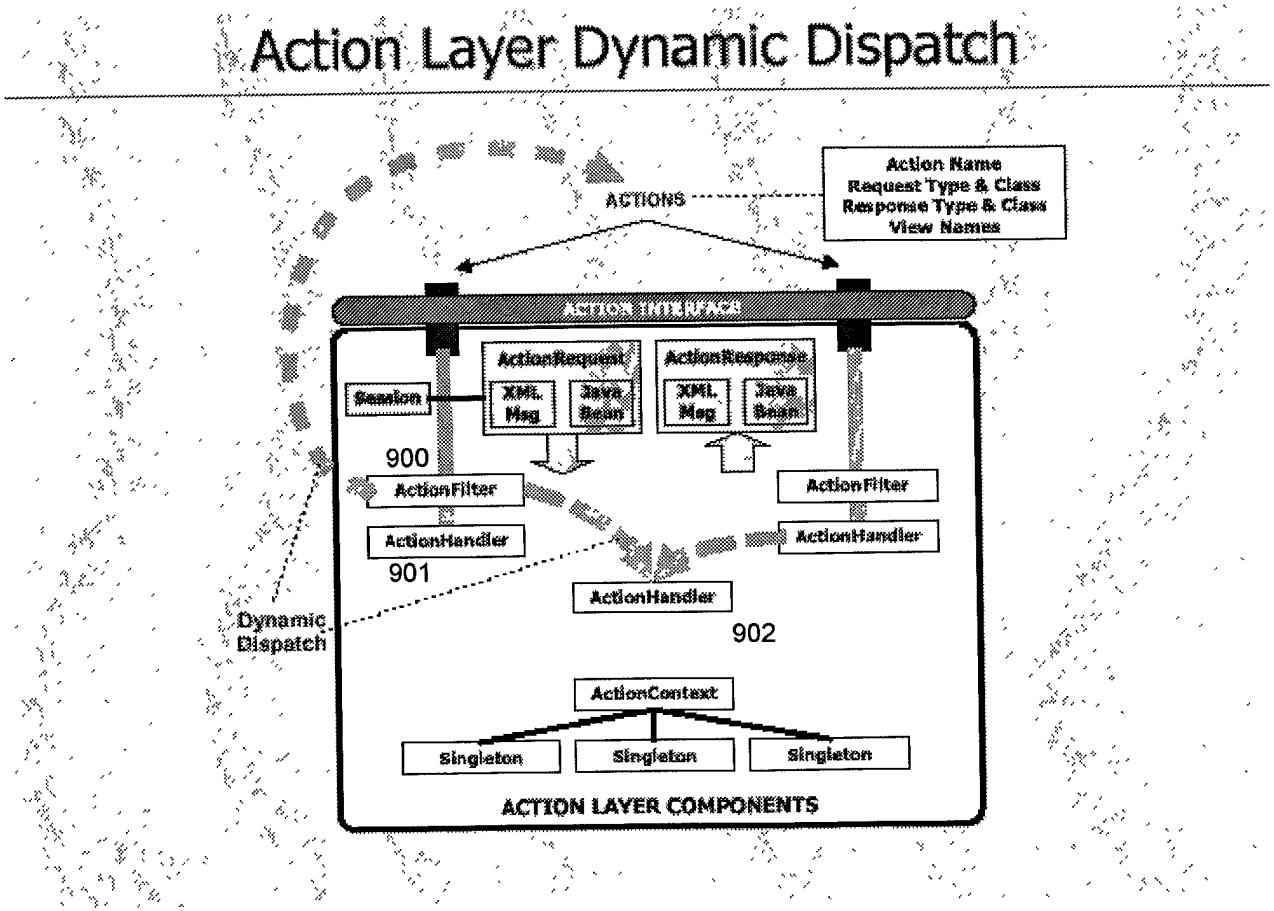

Action Layer Dynamic Dispatch
ACTIONS
Action Name
Request Type & Class
Response Type & Class
View Names
ACTION INTERFACE
ActionRequest
XML Msg
Java Bean
ActionResponse
XML Msg
Java Bean
Session
900
ActionFilter
ActionHandler
901
ActionFilter
ActionHandler
ActionHandler
902
Dynamic Dispatch
ActionContext
Singleton
Singleton
Singleton
ACTION LAYER COMPONENTS

Fig. 9

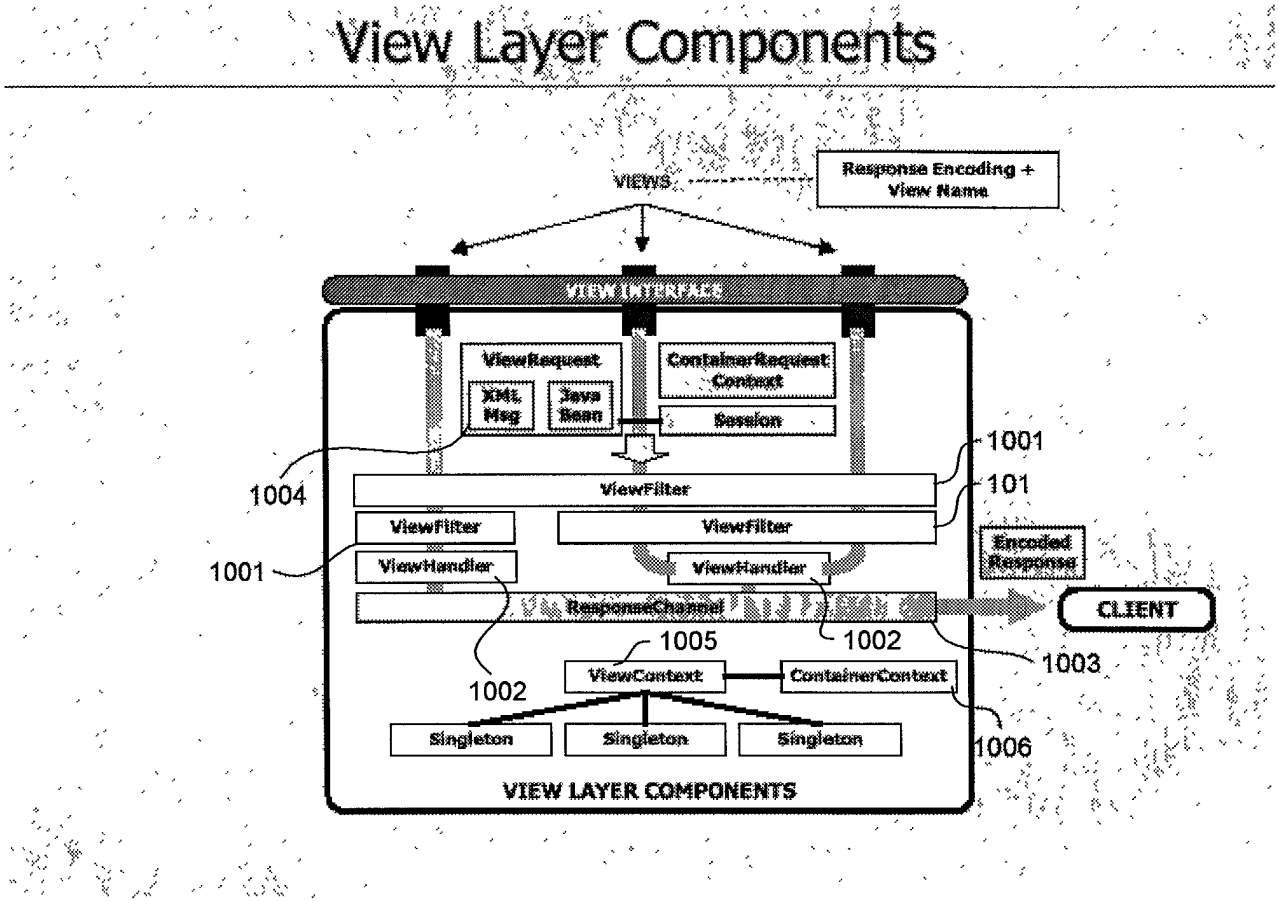

View Layer Components
VIEWS
Response Encoding + View Name
VIEW INTERFACES
ViewRequest
XML Msg
Java Bean
ContainerRequest Context
Session
1004
1001
ViewFilter
ViewFilter
ViewFilter
1001
101
ViewHandler
ViewHandler
Encoded Response
ResponseChannel
CLIENT
1005
1002
1003
1002
ViewContext
ContainerContext
Singleton
Singleton
Singleton
1006
VIEW LAYER COMPONENTS

Fig. 10

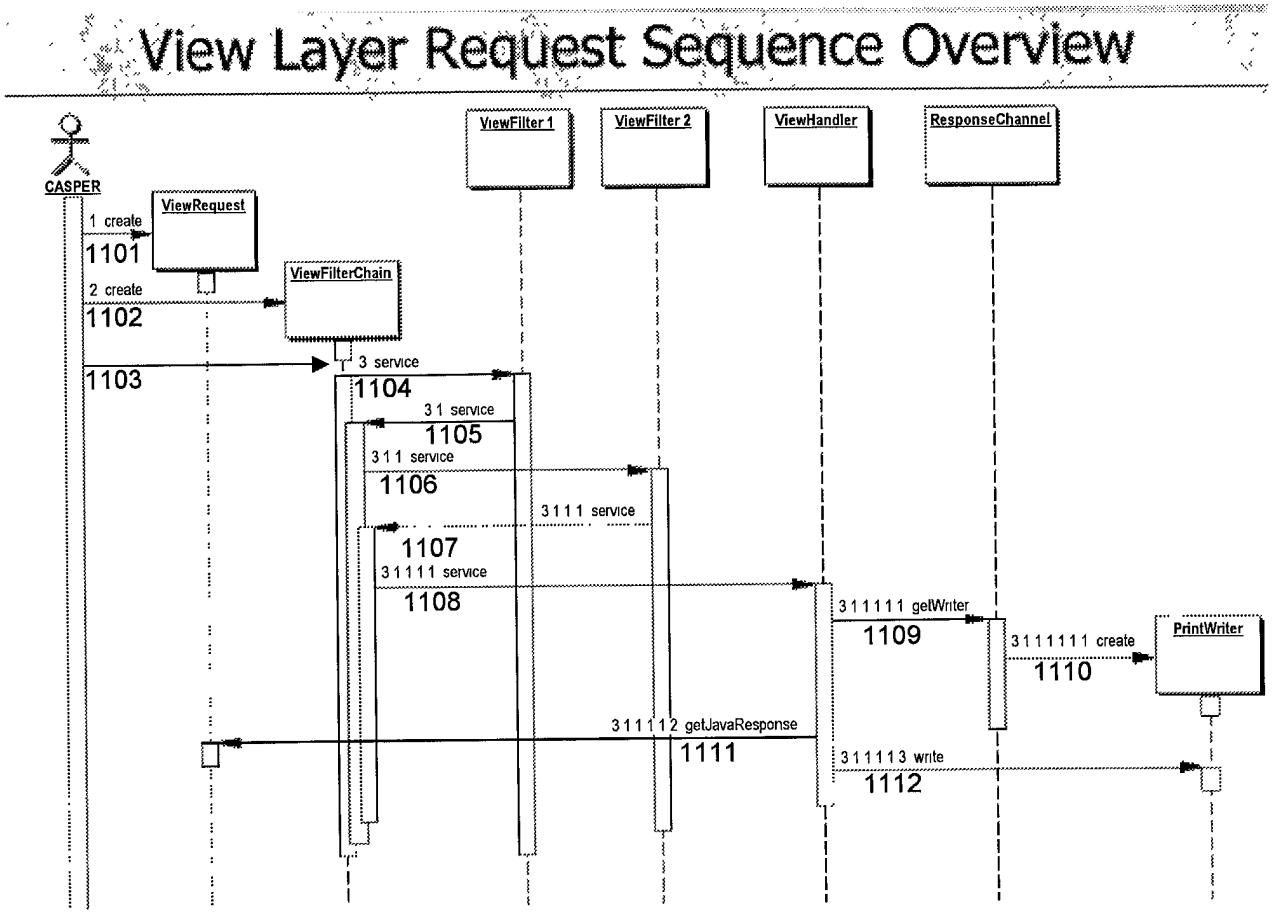

View Layer Request Sequence Overview
CASPER
ViewRequest
ViewFilterChain
ViewFilter 1
ViewFilter 2
ViewHandler
ResponseChannel
PrintWriter
1 create
1101
2 create
1102
1103
3 service
1104
31 service
1105
311 service
1106
3111 service
1107
31111 service
1108
311111 getWriter
1109
3111111 create
1110
311112 getJavaResponse
1111
311113 write
1112

Fig. 11

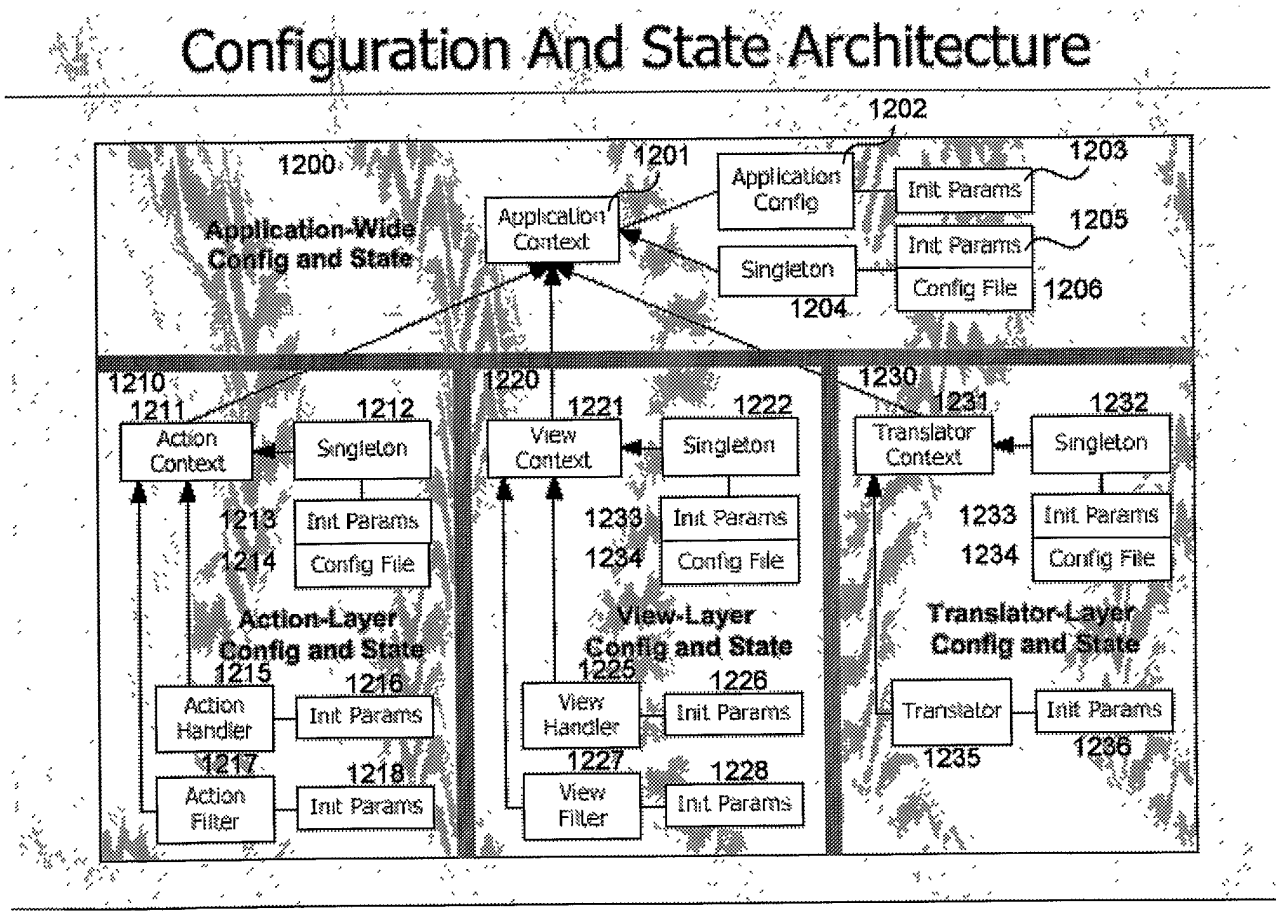

Configuration And State Architecture
1200
Application-Wide Config and State
1201
Application Context
1202
Application Config
1203
Init Params
1205
Init Params
Singleton
1204
Config File
1206
1210
1211
Action Context
1212
Singleton
1213
Init Params
1214
Config File
Action-Layer Config and State
1215
Action Handler
1216
Init Params
1217
Action Filter
1218
Init Params
1220
1221
View Context
1222
Singleton
1233
Init Params
1234
Config File
View-Layer Config and State
1225
View Handler
1226
Init Params
1227
View Filter
1228
Init Params
1230
1231
Translator Context
1232
Singleton
1233
Init Params
1234
Config File
Translator-Layer Config and State
Translator
1235
Init Params
1236

Fig. 12

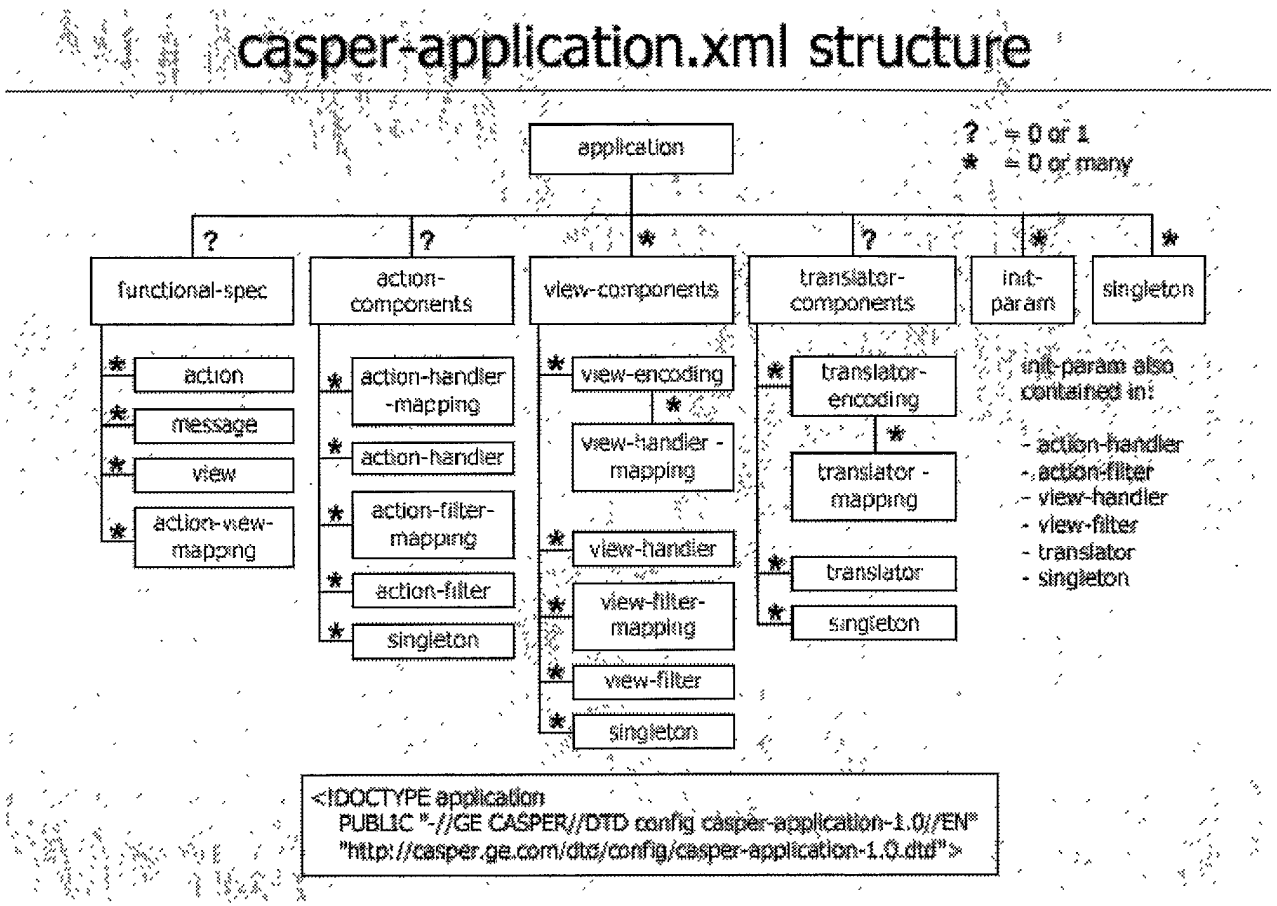

casper-application.xml structure
application
? = 0 or 1
* = 0 or many
functional-spec
action-components
view-components
translator-components
init-param
singleton
action
message
view
action-view-mapping
action-handler-mapping
action-handler
action-filter-mapping
action-filter
singleton
view-encoding
view-handler-mapping
view-handler
view-filter-mapping
view-filter
singleton
translator-encoding
translator-mapping
translator
singleton
init-param also contained in:
- action-handler
- action-filter
- view-handler
- view-filter
- translator
- singleton
<!DOCTYPE application
PUBLIC "-//GE CASPER//DTD config casper-application-1.0//EN"
"http://casper.ge.com/dtd/config/casper-application-1.0.dtd">>

Fig. 13

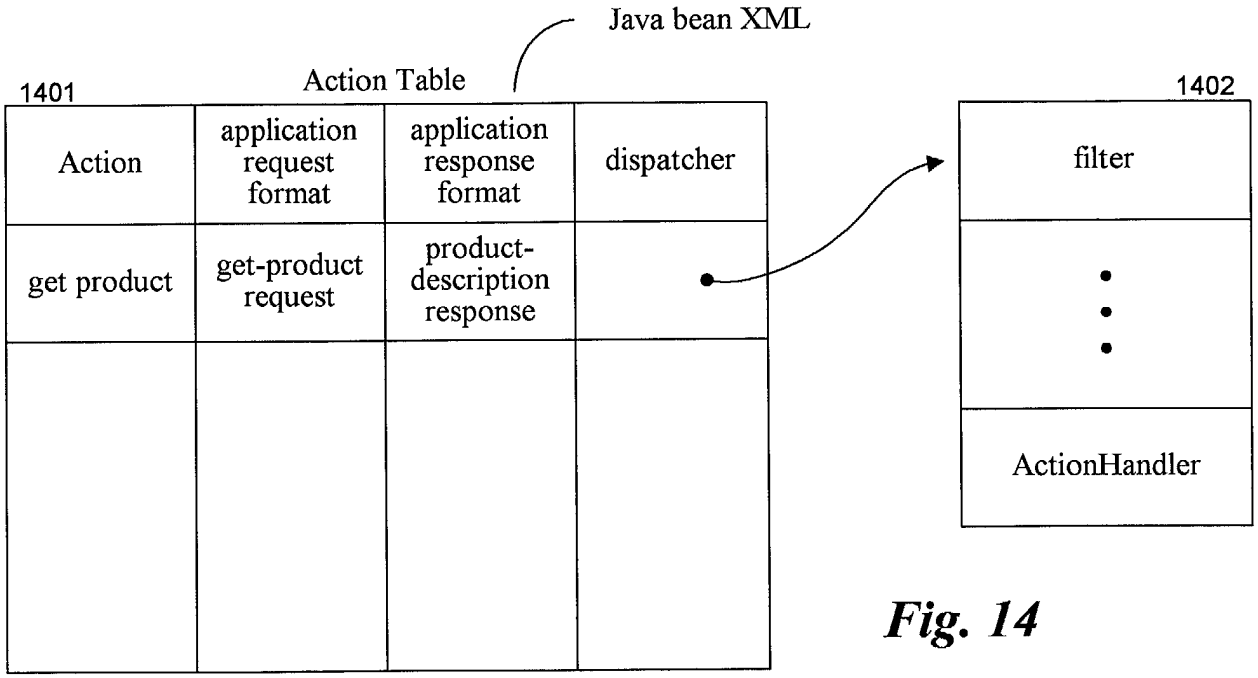

Java bean XML
1401
Action Table
Action
application request format
application response format
dispatcher
get product
get-product request
product-description response
1402
filter
ActionHandler

Fig. 14

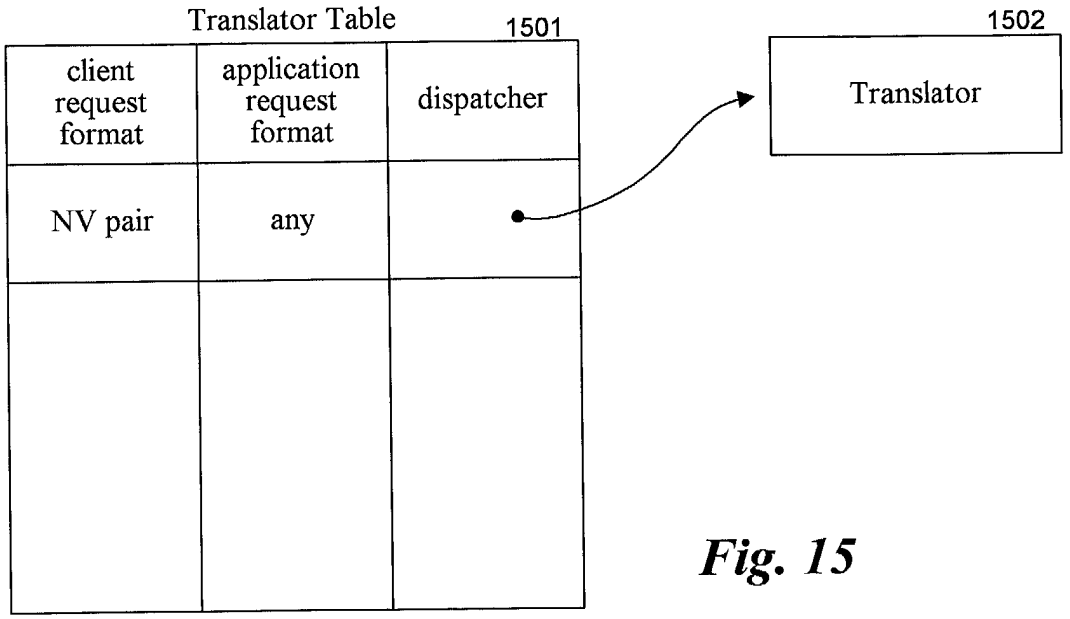

Translator Table
1501
client request format
application request format
dispatcher
NV pair
any
1502
Translator

Fig. 15

TOTP90 26555260

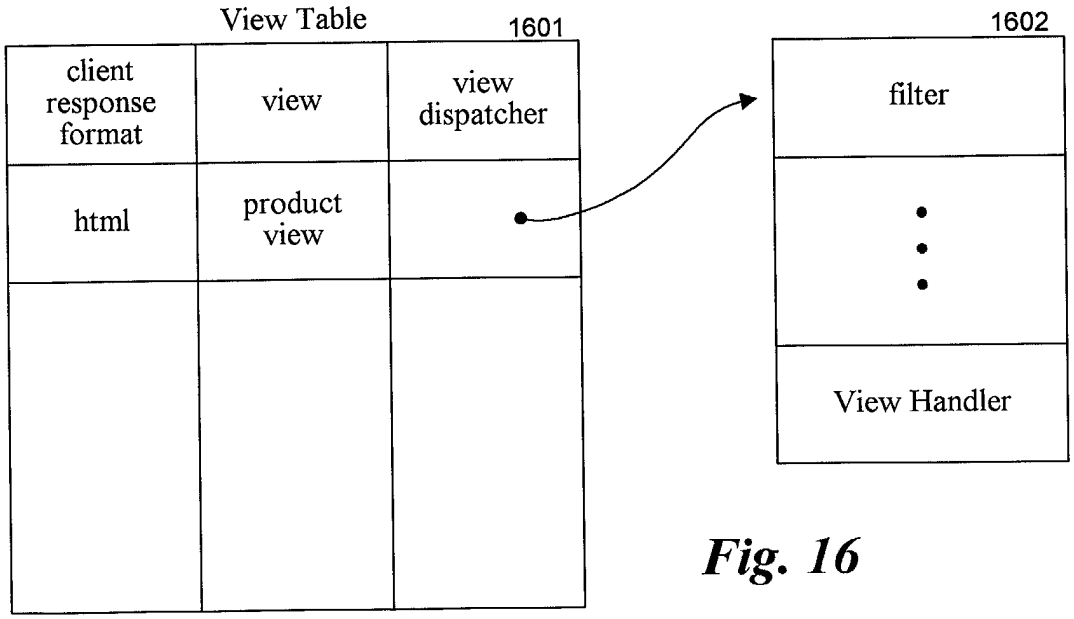


Fig. 16

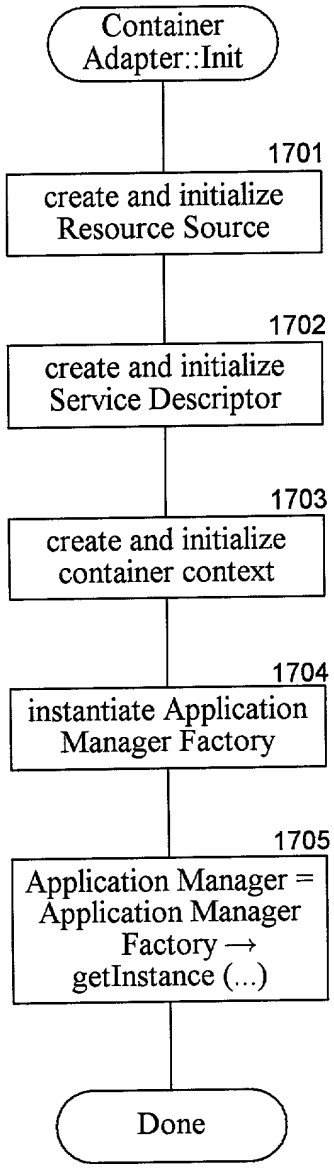

Container Adapter::Init
1701
create and initialize Resource Source
1702
create and initialize Service Descriptor
1703
create and initialize container context
1704
instantiate Application Manager Factory
1705
Application Manager = Application Manager Factory → getInstance (...)
Done

Fig. 17

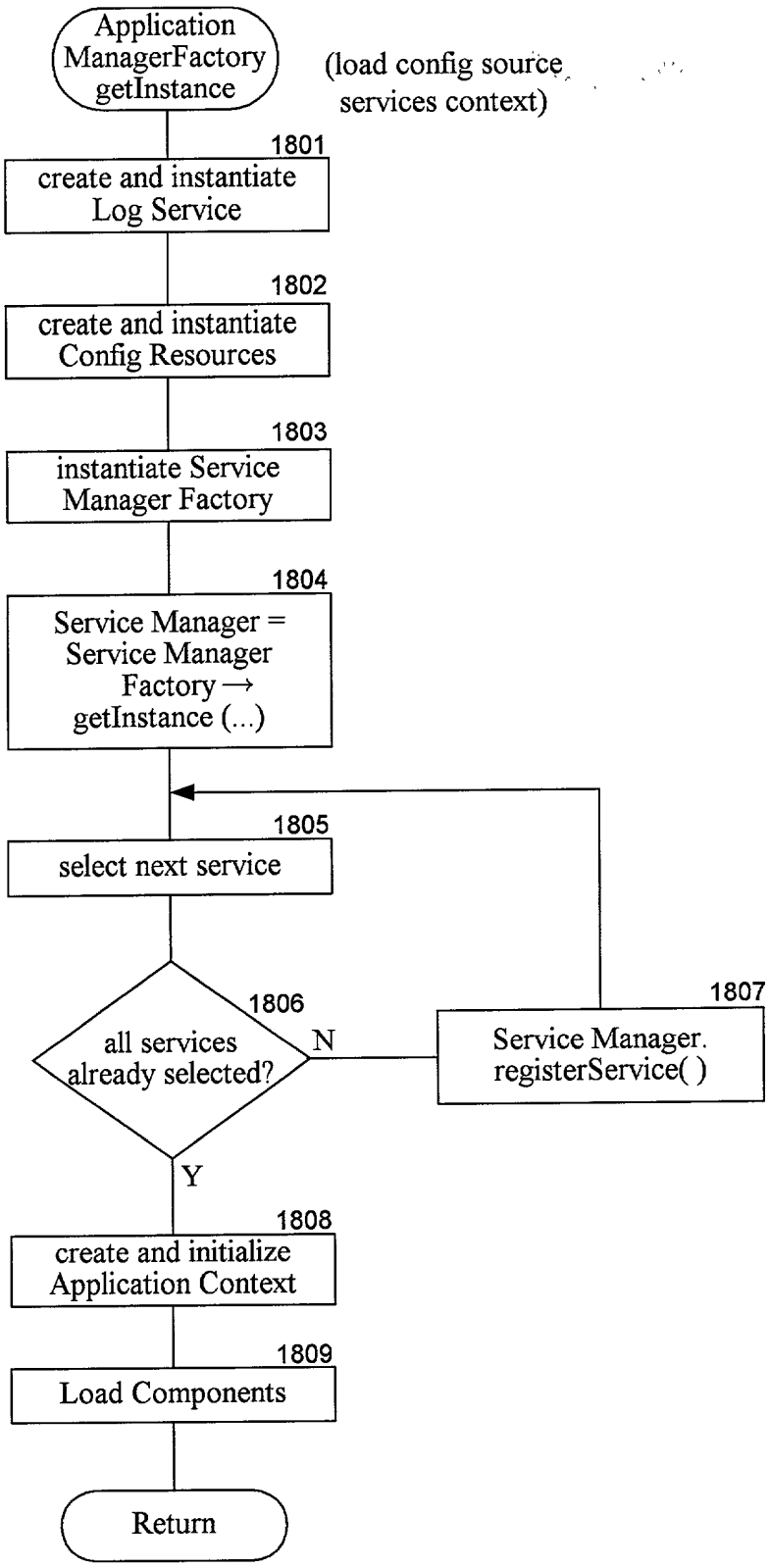

Application ManagerFactory getInstance
(load config source services context)
1801
create and instantiate Log Service
1802
create and instantiate Config Resources
1803
instantiate Service Manager Factory
1804
Service Manager = Service Manager Factory → getInstance (...)
1805
select next service
1806
all services already selected?
N
1807
Service Manager. registerService( )
Y
1808
create and initialize Application Context
1809
Load Components
Return

Fig. 18

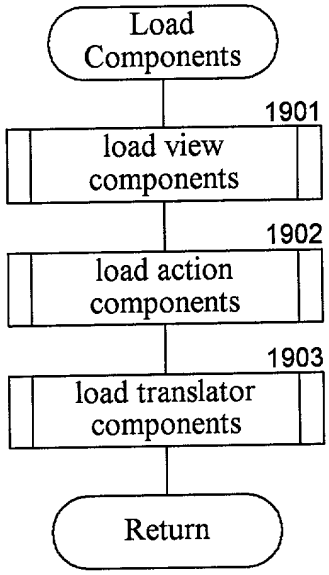

Load Components
1901
load view components
1902
load action components
1903
load translator components
Return

Fig. 19

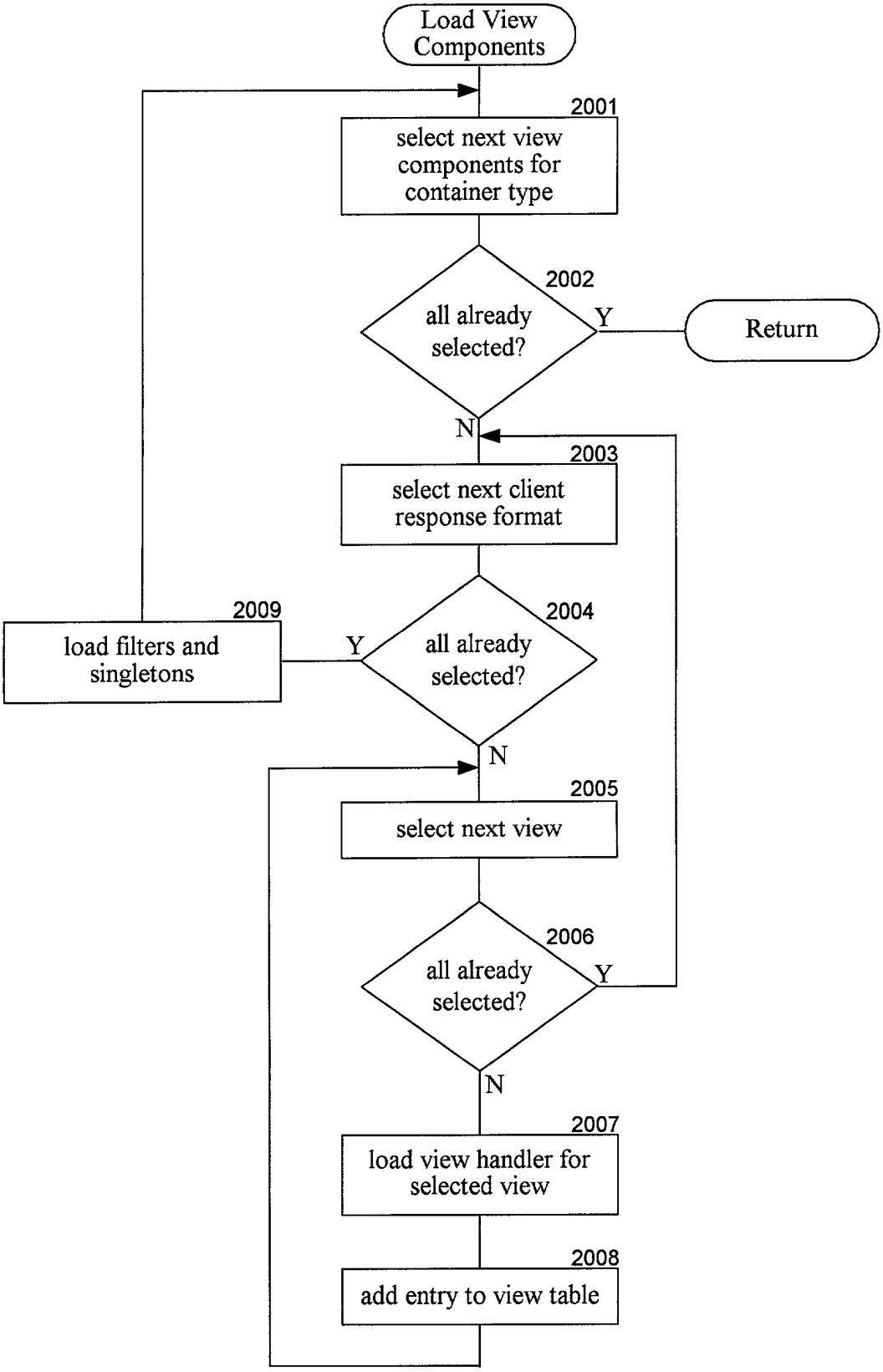

Load View Components
2001
select next view components for container type
2002
all already selected?
Y
Return
N
2003
select next client response format
2004
all already selected?
Y
2009
load filters and singletons
N
2005
select next view
2006
all already selected?
Y
N
2007
load view handler for selected view
2008
add entry to view table

Fig. 20

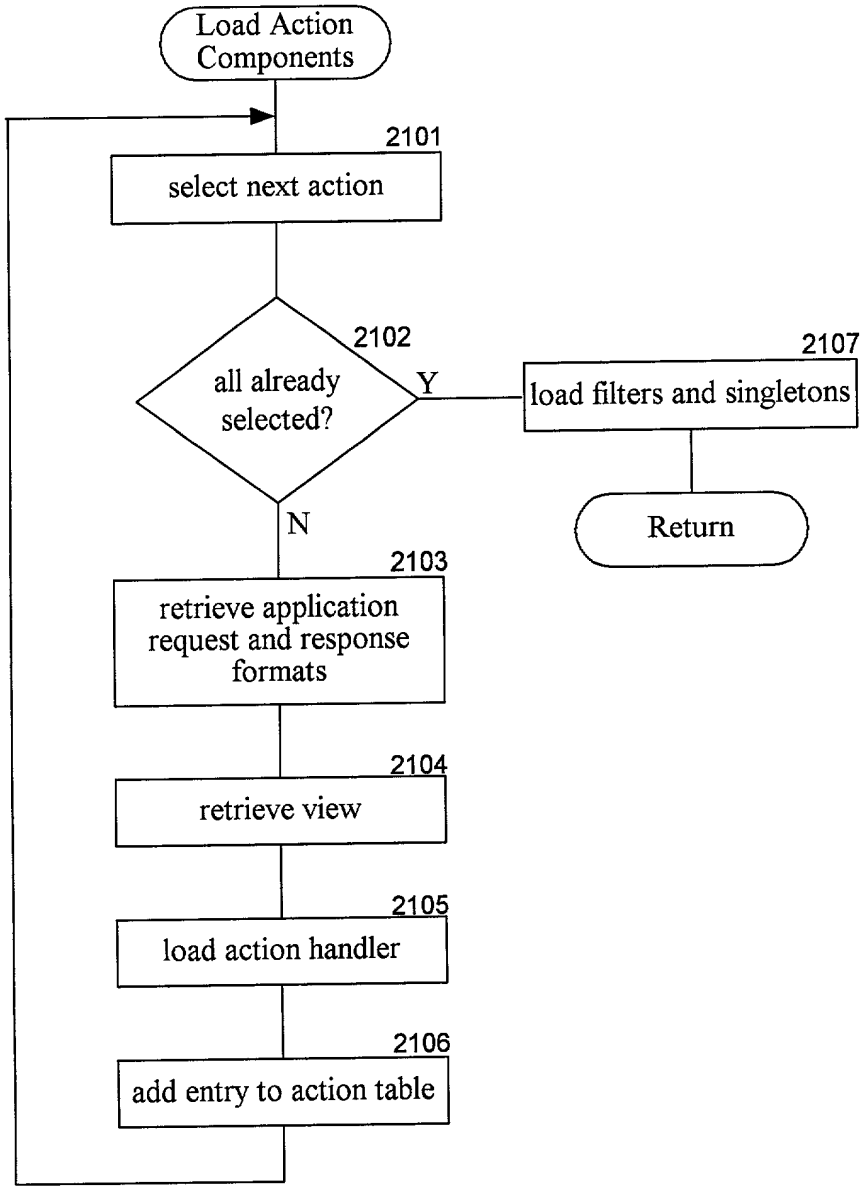

Load Action Components
2101
select next action
2102
all already selected?
Y
2107
load filters and singletons
Return
N
2103
retrieve application request and response formats
2104
retrieve view
2105
load action handler
2106
add entry to action table

Fig. 21

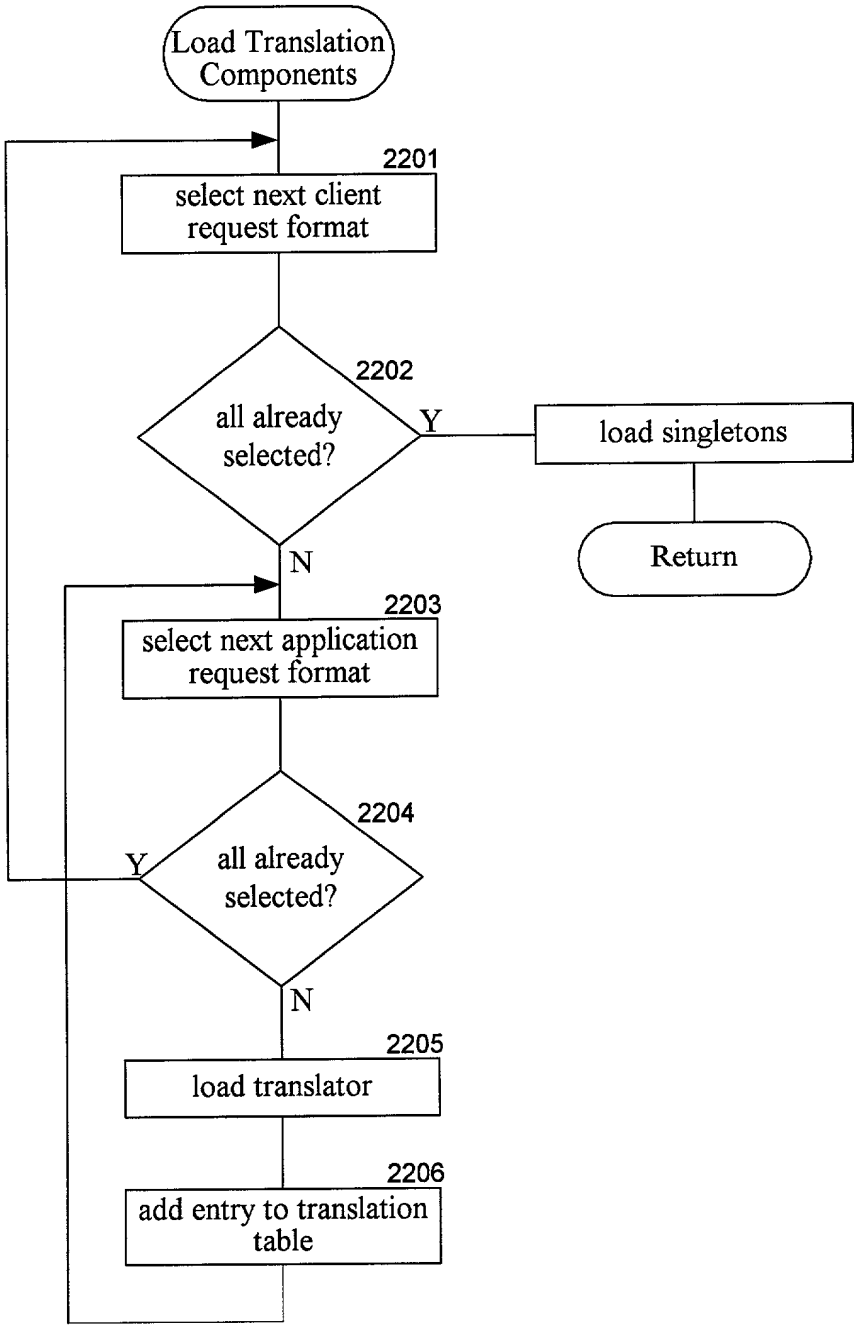

Load Translation Components
2201
select next client request format
2202
all already selected?
Y
load singletons
Return
N
2203
select next application request format
2204
all already selected?
Y
N
2205
load translator
2206
add entry to translation table

Fig. 22

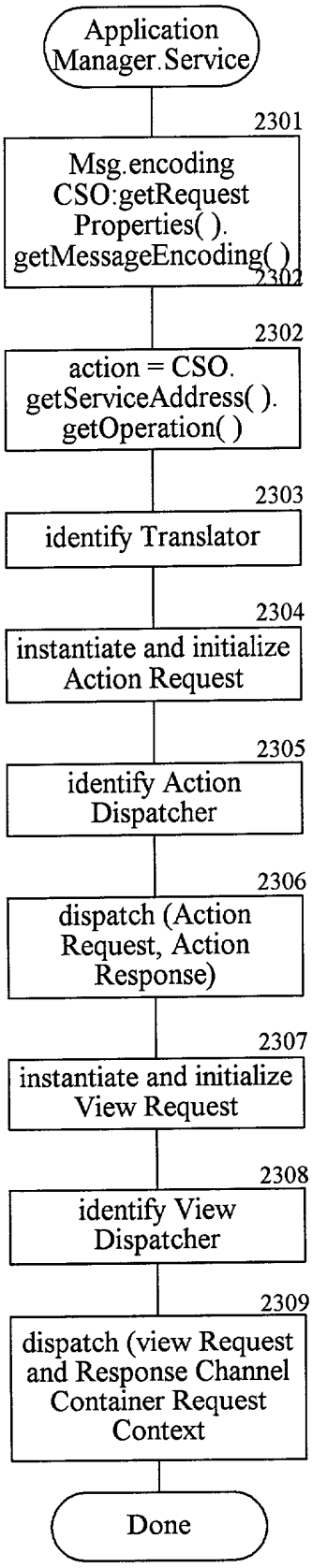

Application Manager.Service
(Container Service Order)
2301
Msg.encoding CSO:getRequest Properties( ). getMessageEncoding( )
2302
action = CSO. getServiceAddress( ). getOperation( )
2303
identify Translator
2304
instantiate and initialize Action Request
2305
identify Action Dispatcher
2306
dispatch (Action Request, Action Response)
2307
instantiate and initialize View Request
2308
identify View Dispatcher
2309
dispatch (view Request and Response Channel Container Request Context
Done

Fig. 23

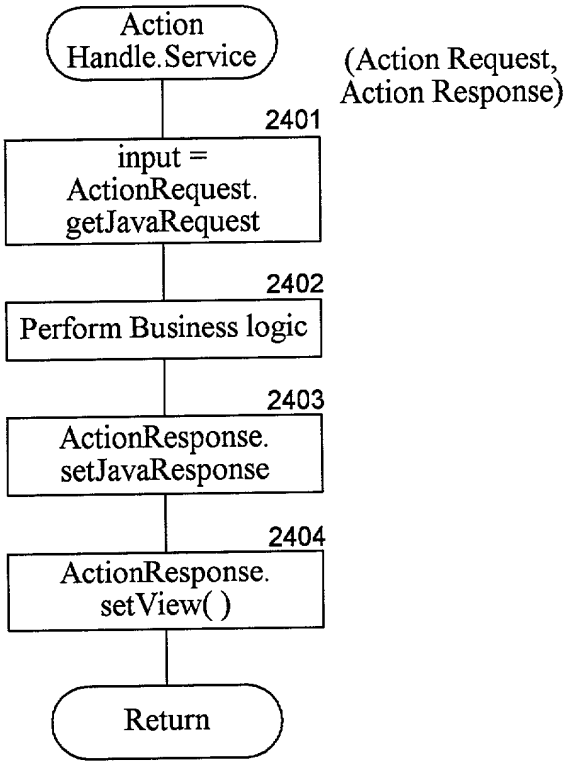

Action
Handle.Service
(Action Request,
Action Response)
2401
input =
ActionRequest.
getJavaRequest
2402
Perform Business logic
2403
ActionResponse.
setJavaResponse
2404
ActionResponse.
setView( )
Return

Fig. 24

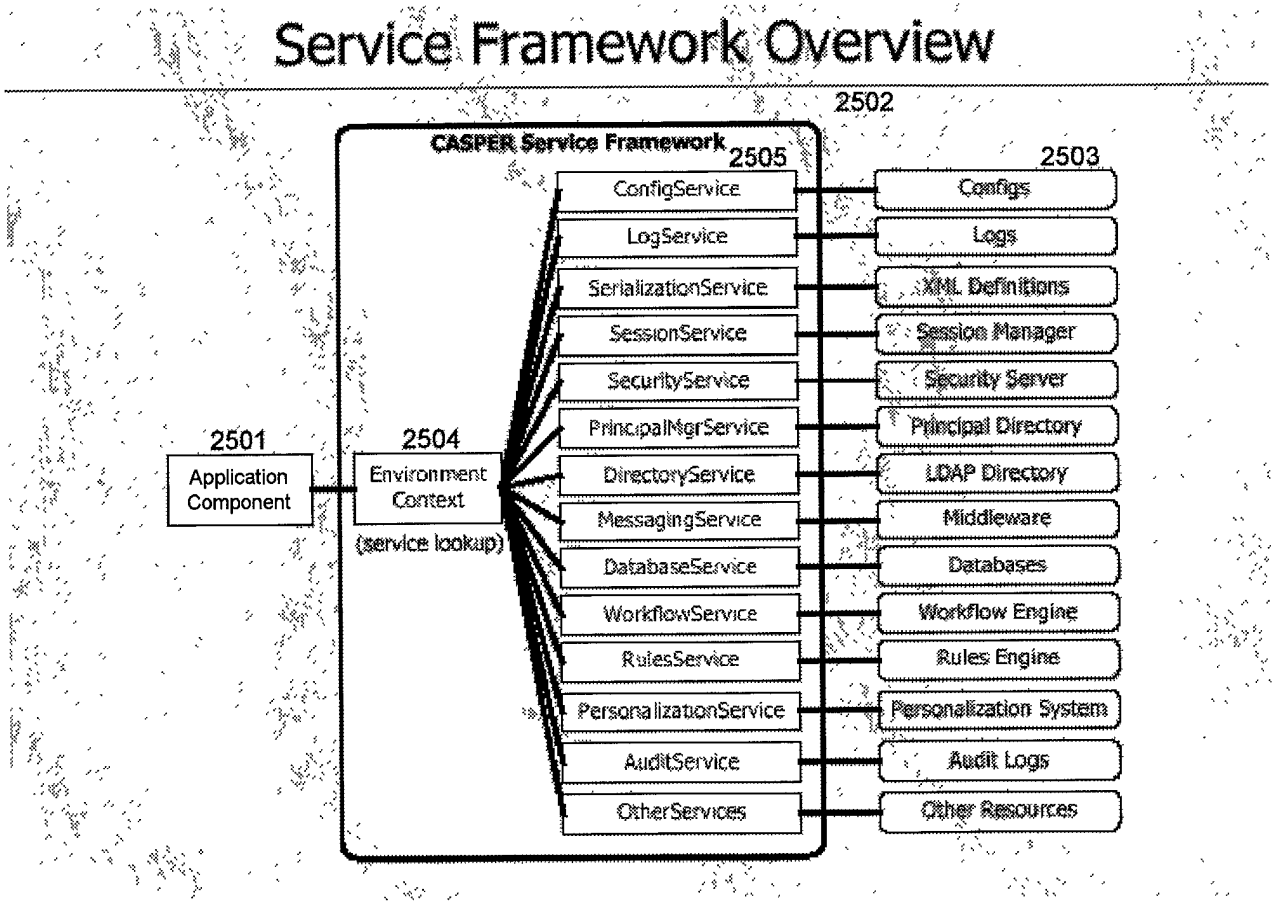

Service Framework Overview
2502
CASPER Service Framework
2505
2503
ConfigService
LogService
SerializationService
SessionService
SecurityService
PrincipalMgrService
DirectoryService
MessagingService
DatabaseService
WorkflowService
RulesService
PersonalizationService
AuditService
OtherServices
Configs
Logs
XML Definitions
Session Manager
Security Server
Principal Directory
LDAP Directory
Middleware
Databases
Workflow Engine
Rules Engine
Personalization System
Audit Logs
Other Resources
2501
Application Component
2504
Environment Context
(service lookup)

Fig. 25

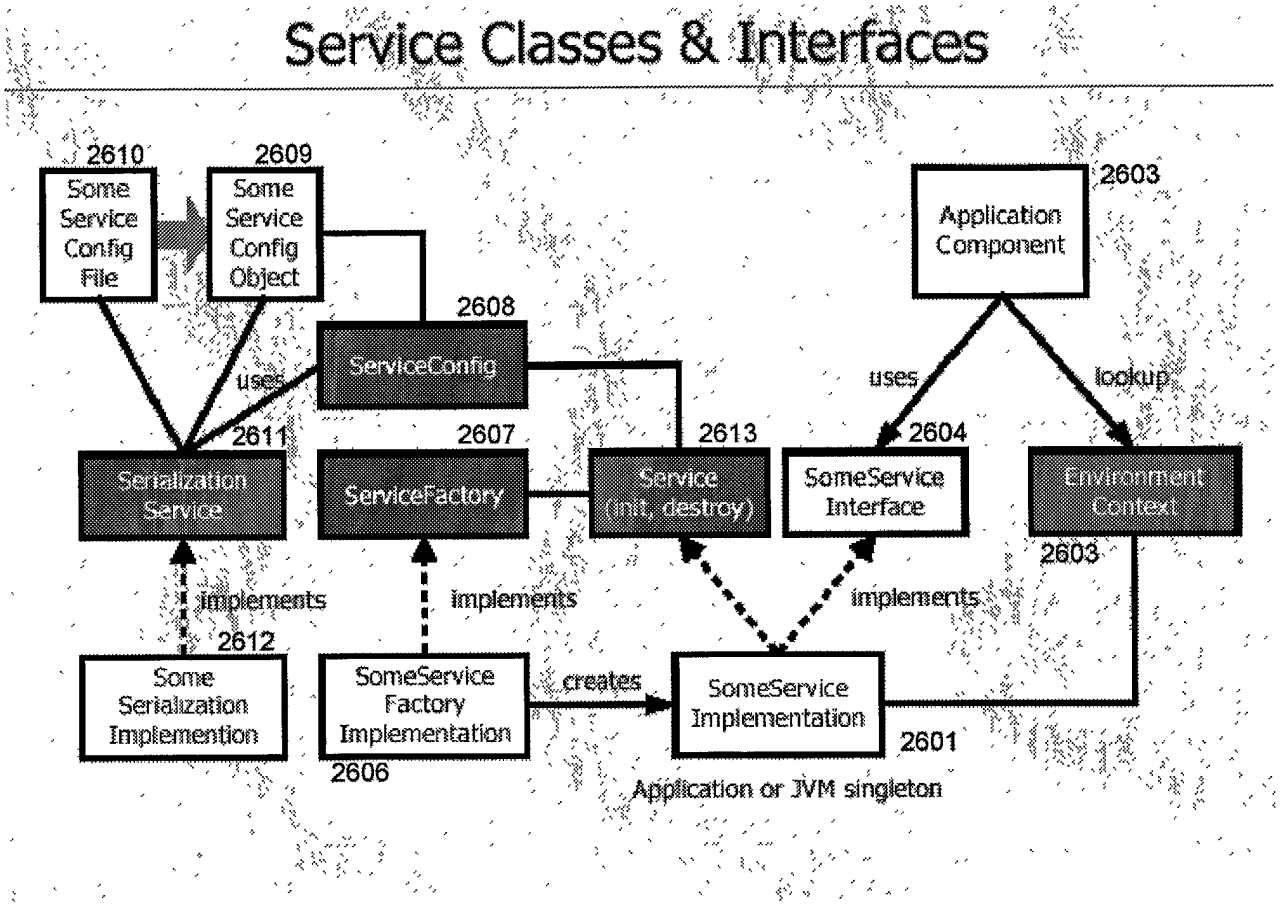

Service Classes & Interfaces
2610
Some Service Config File
2609
Some Service Config Object
2608
ServiceConfig
uses
2611
Serialization Service
2607
ServiceFactory
2613
Service (init, destroy)
2603
Application Component
uses
2604
SomeService Interface
lookup
Environment Context
2603
implements
2612
Some Serialization Implemention
implements
SomeService Factory Implementation
2606
creates
SomeService Implementation
2601
implements
Application or JVM singleton

Fig. 26

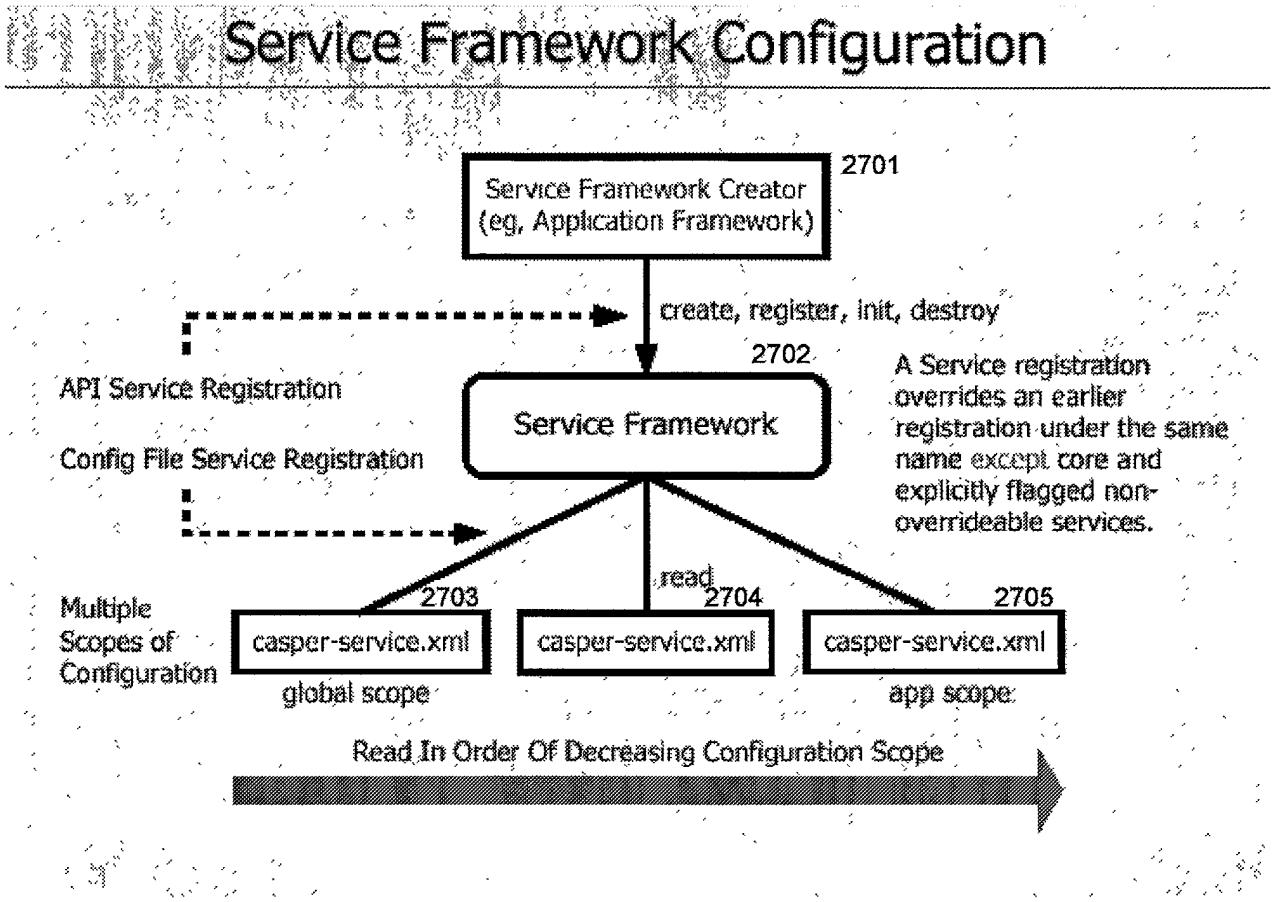

Service Framework Configuration
Service Framework Creator
(eg, Application Framework)
2701
create, register, init, destroy
2702
Service Framework
API Service Registration
Config File Service Registration
A Service registration overrides an earlier registration under the same name except core and explicitly flagged non-overrideable services.
read
Multiple Scopes of Configuration
2703
casper-service.xml
global scope
2704
casper-service.xml
2705
casper-service.xml
app scope
Read In Order Of Decreasing Configuration Scope

Fig. 27

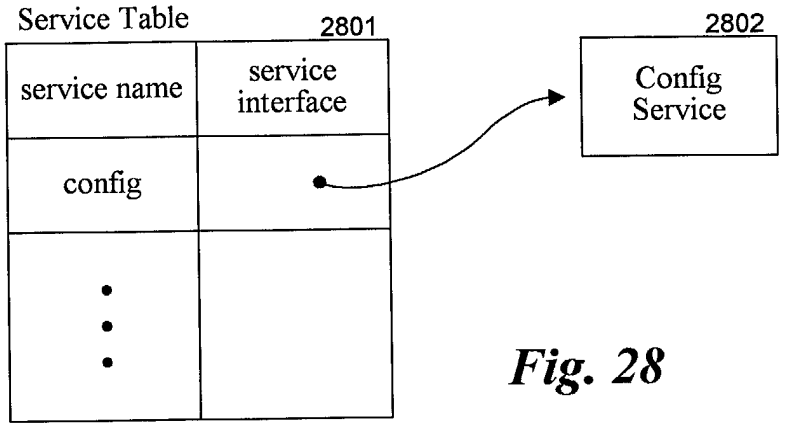

Service Table
2801
service name
service interface
config
2802
Config Service

Fig. 28

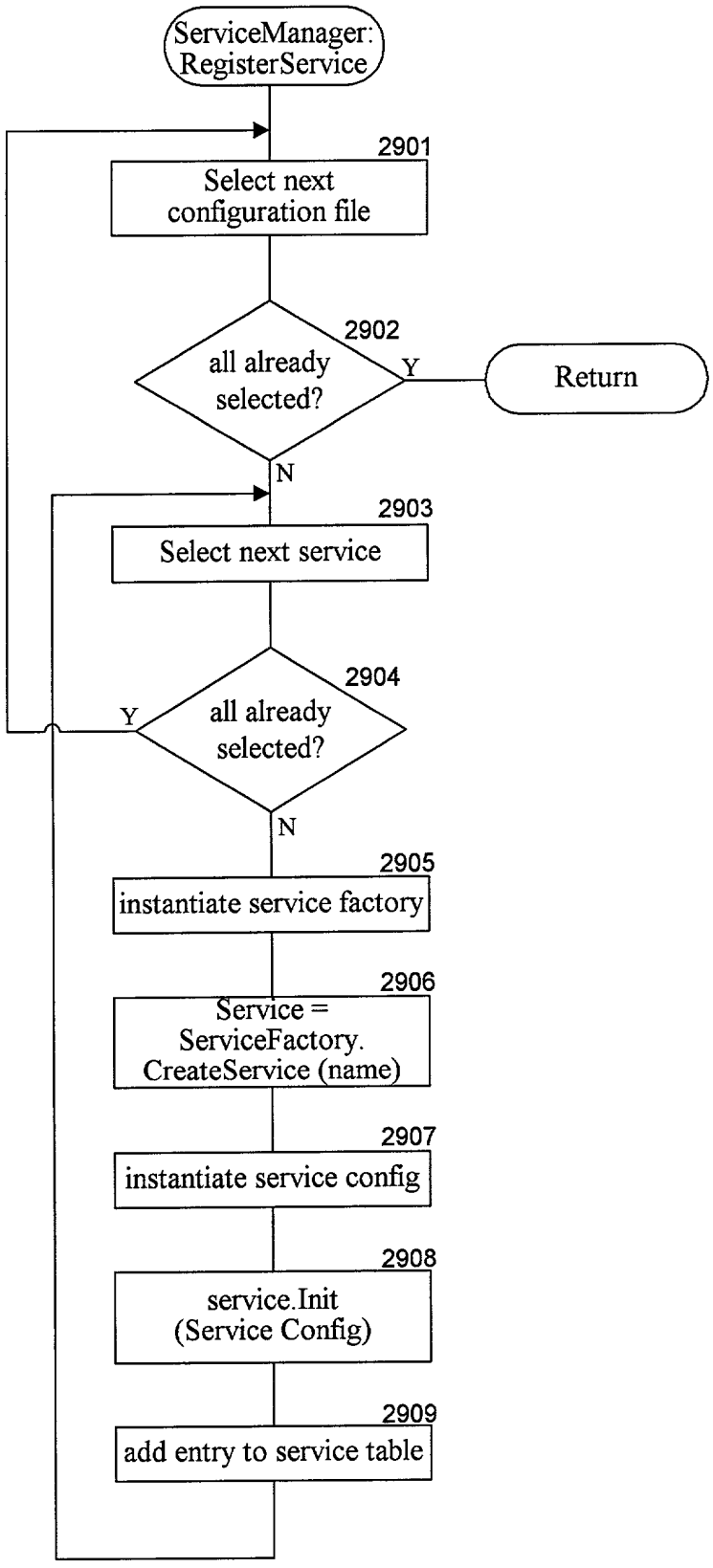

ServiceManager: RegisterService
2901
Select next configuration file
2902
all already selected?
Y
Return
N
2903
Select next service
2904
all already selected?
Y
N
2905
instantiate service factory
2906
Service = ServiceFactory. CreateService (name)
2907
instantiate service config
2908
service.Init (Service Config)
2909
add entry to service table

Fig. 29

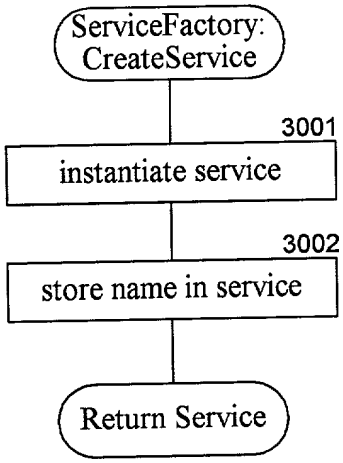

ServiceFactory:
CreateService
(name service)
3001
instantiate service
3002
store name in service
Return Service

Fig. 30

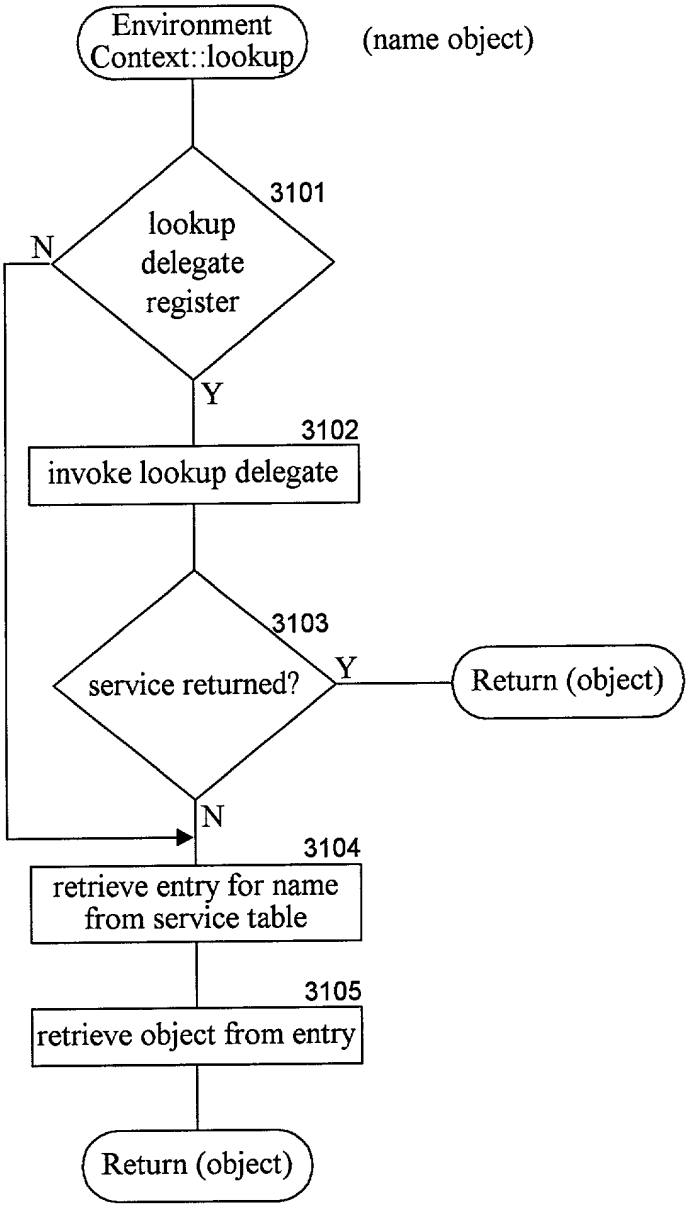

Environment Context::lookup
(name object)
3101
lookup delegate register
N
Y
3102
invoke lookup delegate
3103
service returned?
Y
Return (object)
N
3104
retrieve entry for name from service table
3105
retrieve object from entry
Return (object)

Fig. 31

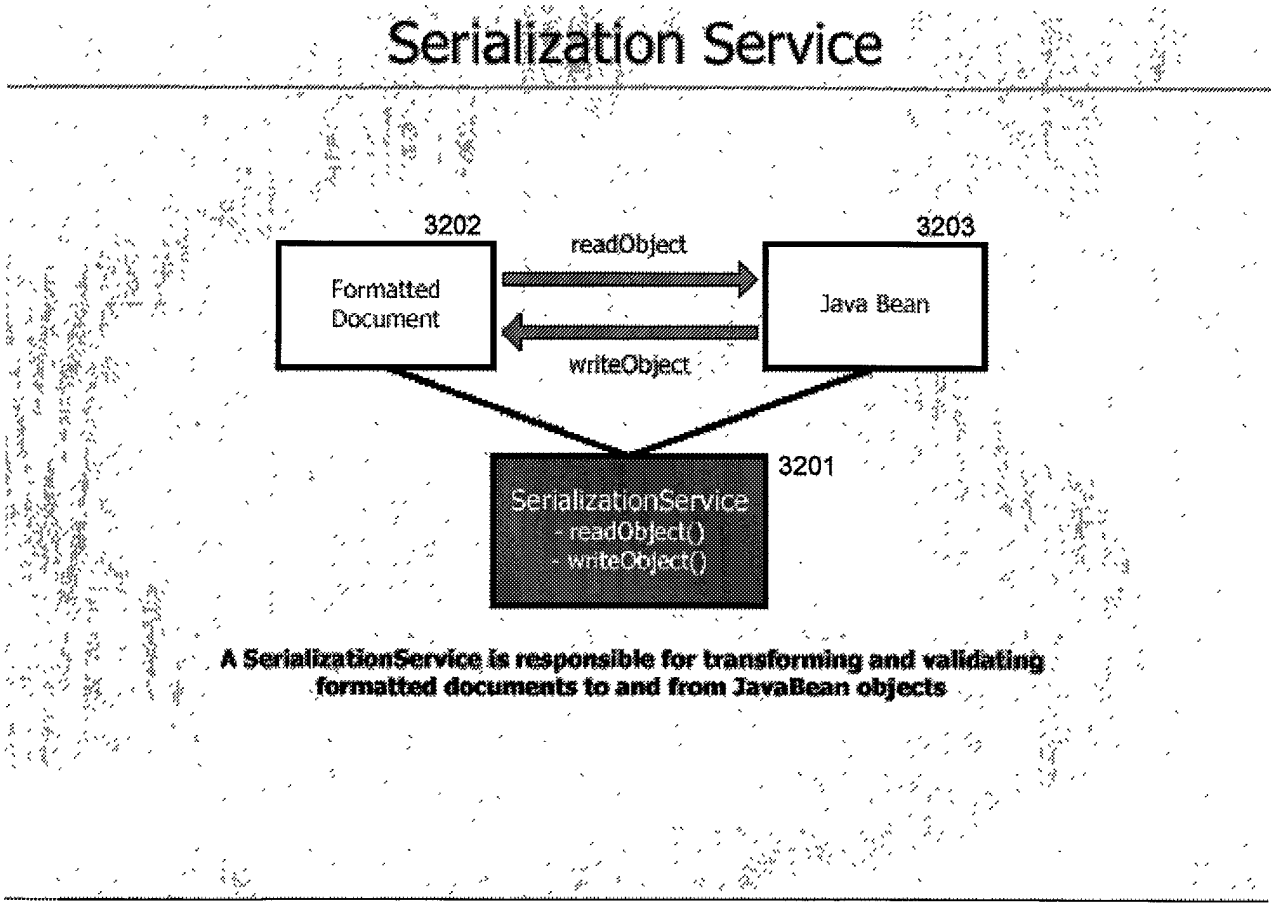

Serialization Service
3202
readObject
3203
Formatted Document
Java Bean
writeObject
3201
SerializationService
- readObject()
- writeObject()
A SerializationService is responsible for transforming and validating formatted documents to and from JavaBean objects

Fig. 32

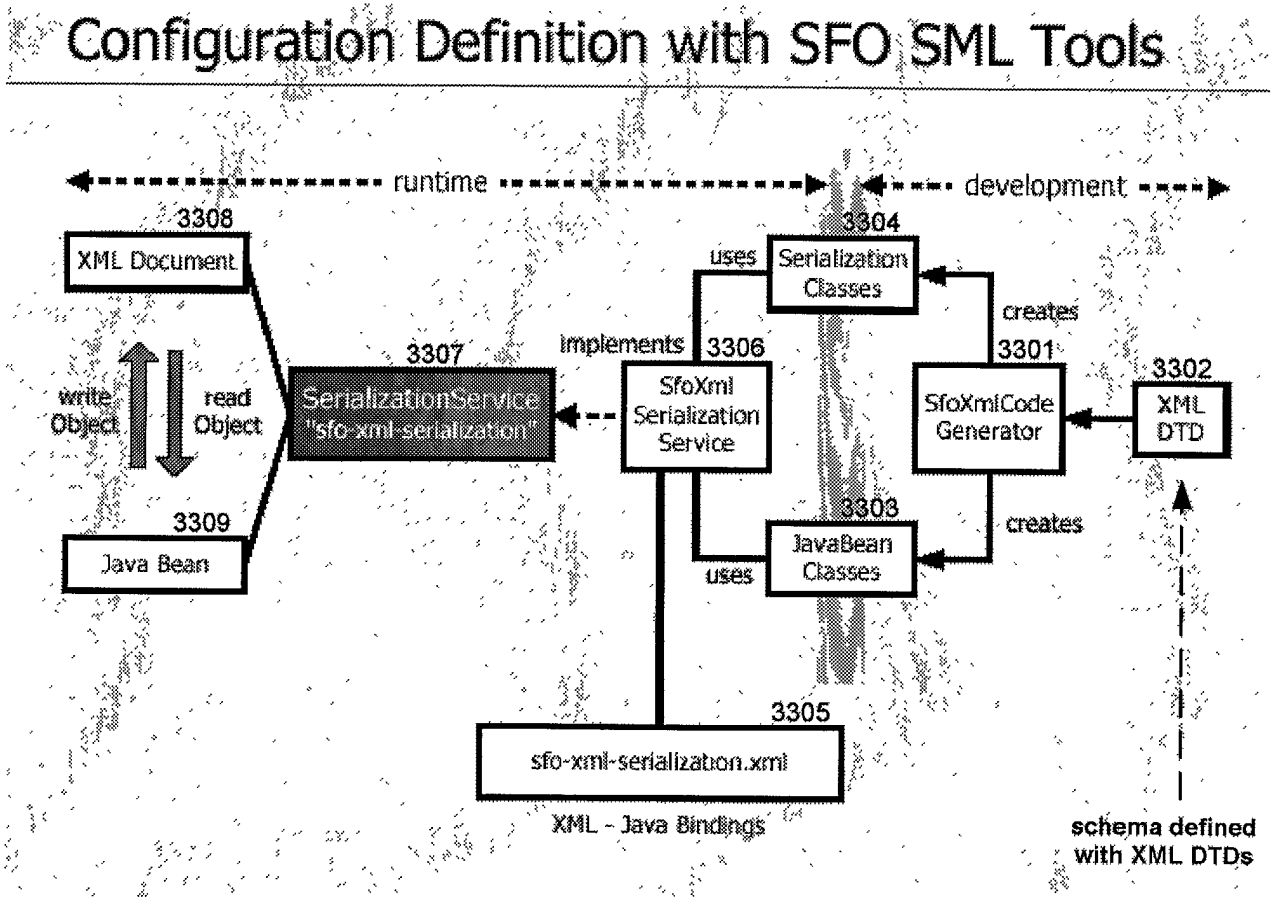

Configuration Definition with SFO SML Tools
runtime
development
3308
XML Document
write Object
read Object
3309
Java Bean
3307
SerializationService "sfo-xml-serialization"
implements
3306
SfoXml Serialization Service
uses
3304
Serialization Classes
creates
3301
SfoXmlCode Generator
3302
XML DTD
3303
JavaBean Classes
uses
creates
3305
sfo-xml-serialization.xml
XML - Java Bindings
schema defined with XML DTDs

Fig. 33

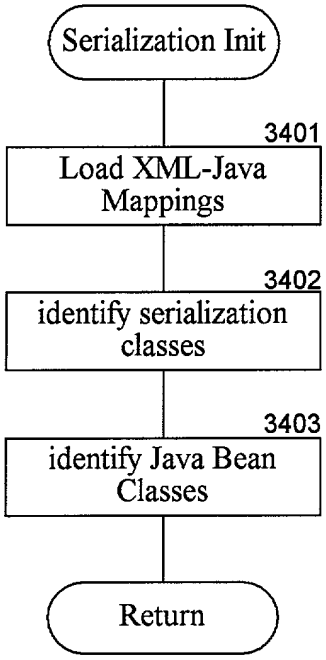

Serialization Init
3401
Load XML-Java Mappings
3402
identify serialization classes
3403
identify Java Bean Classes
Return

Fig. 34

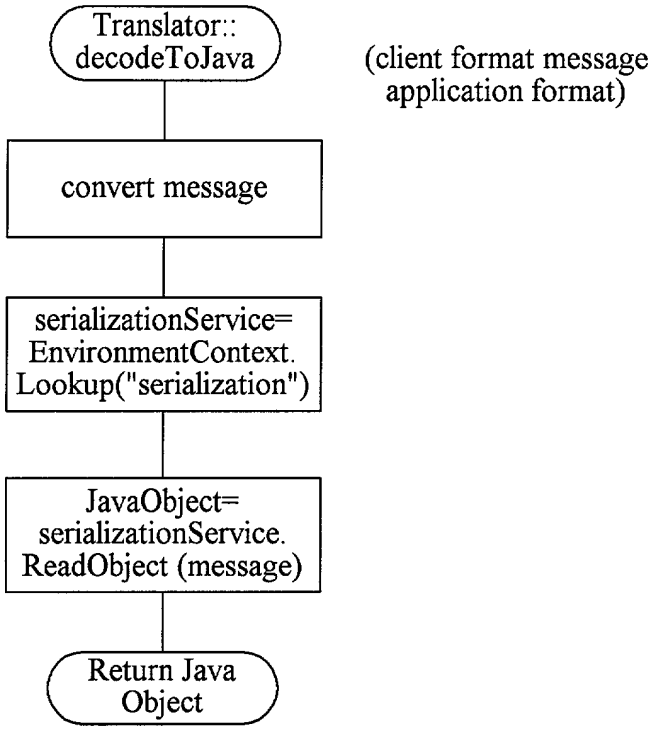

Translator::
decodeToJava
(client format message
application format)
convert message
serializationService=
EnvironmentContext.
Lookup("serialization")
JavaObject=
serializationService.
ReadObject (message)
Return Java
Object

Fig. 35

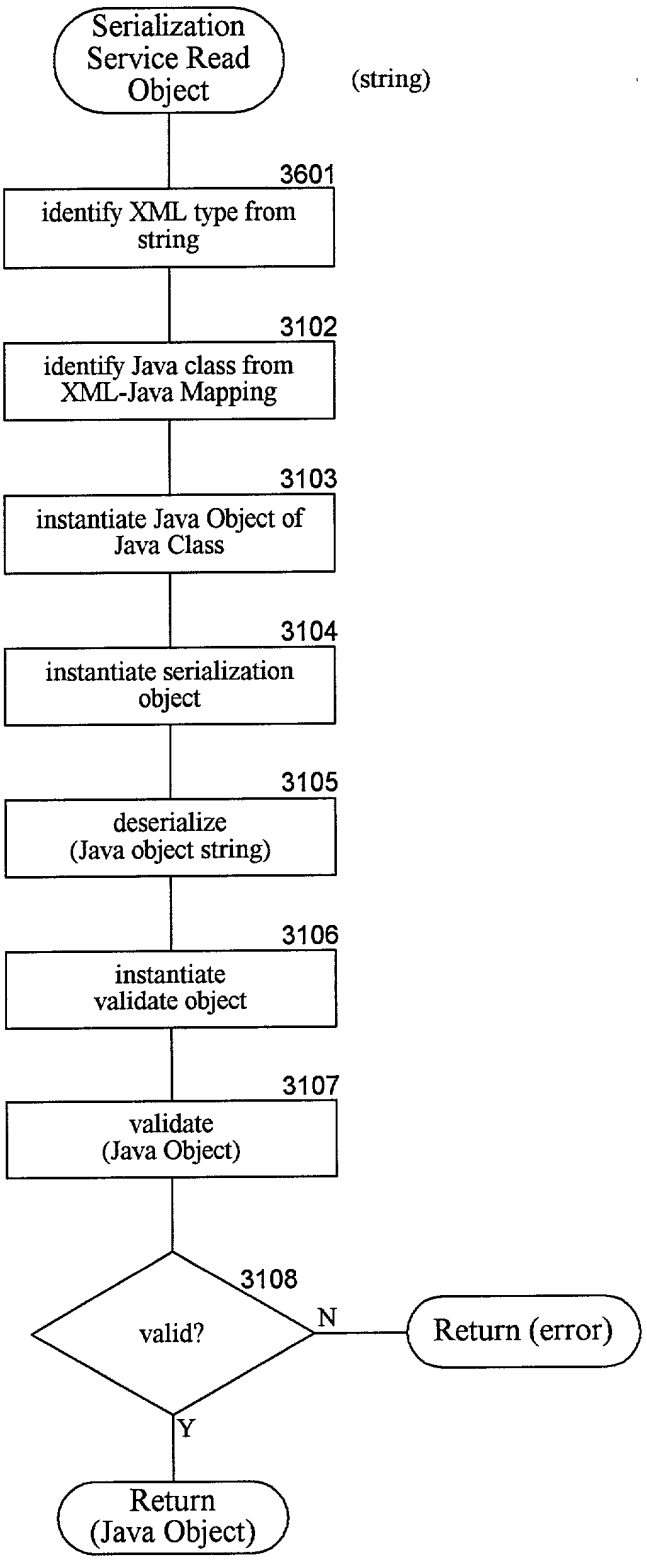

Serialization Service Read Object
(string)
3601
identify XML type from string
3102
identify Java class from XML-Java Mapping
3103
instantiate Java Object of Java Class
3104
instantiate serialization object
3105
deserialize (Java object string)
3106
instantiate validate object
3107
validate (Java Object)
3108
valid?
N
Return (error)
Y
Return (Java Object)

Fig. 36

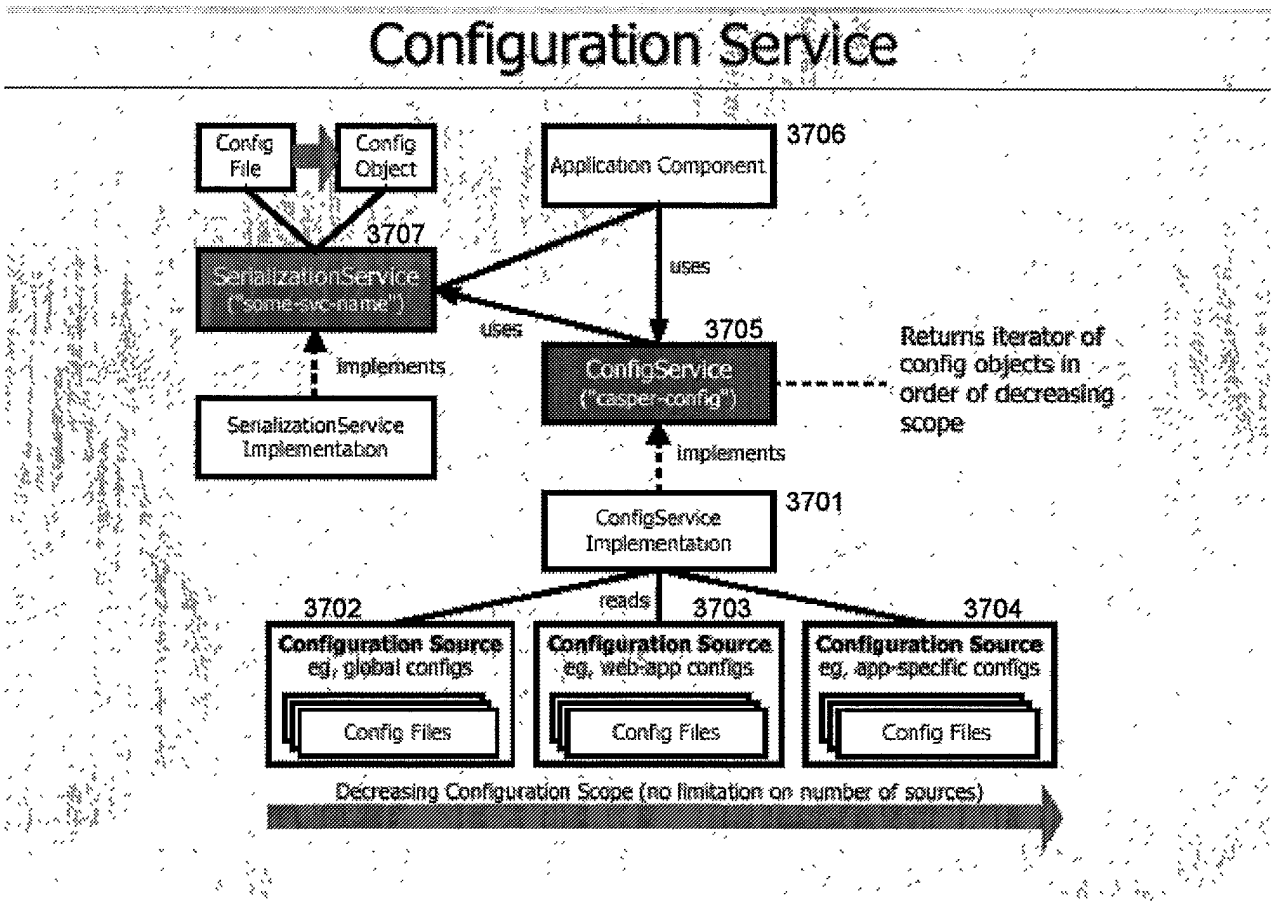

Configuration Service
Config File
Config Object
3707
SerializationService ("some-svc-name")
Implements
SerializationService Implementation
Application Component
3706
uses
uses
3705
ConfigService ("casper-config")
Returns iterator of config objects in order of decreasing scope
implements
ConfigService Implementation
3701
reads
3702
Configuration Source eg, global configs
Config Files
3703
Configuration Source eg, web-app configs
Config Files
3704
Configuration Source eg, app-specific configs
Config Files
Decreasing Configuration Scope (no limitation on number of sources)

Fig. 37

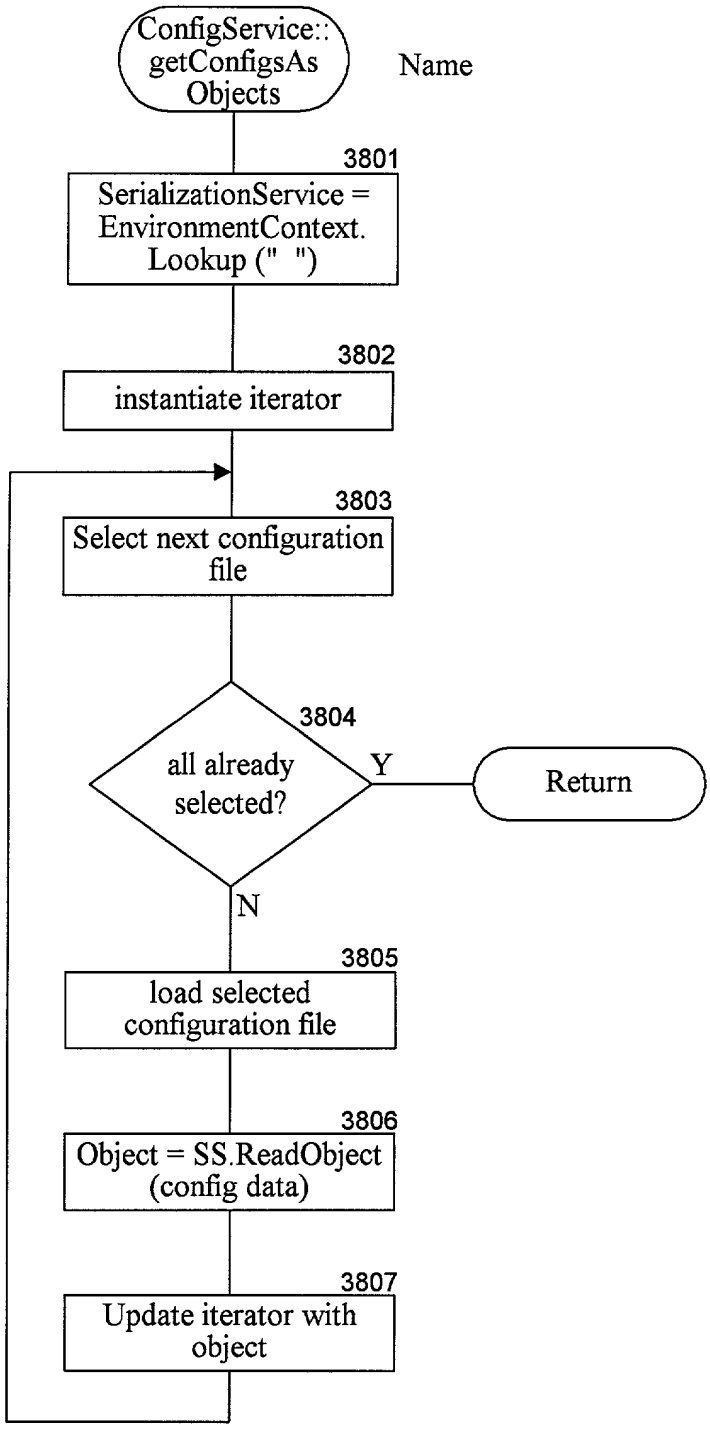

ConfigService::
getConfigsAs
Objects
Name
3801
SerializationService =
EnvironmentContext.
Lookup (" ")
3802
instantiate iterator
3803
Select next configuration
file
3804
all already
selected?
Y
Return
N
3805
load selected
configuration file
3806
Object = SS.ReadObject
(config data)
3807
Update iterator with
object

Fig. 38

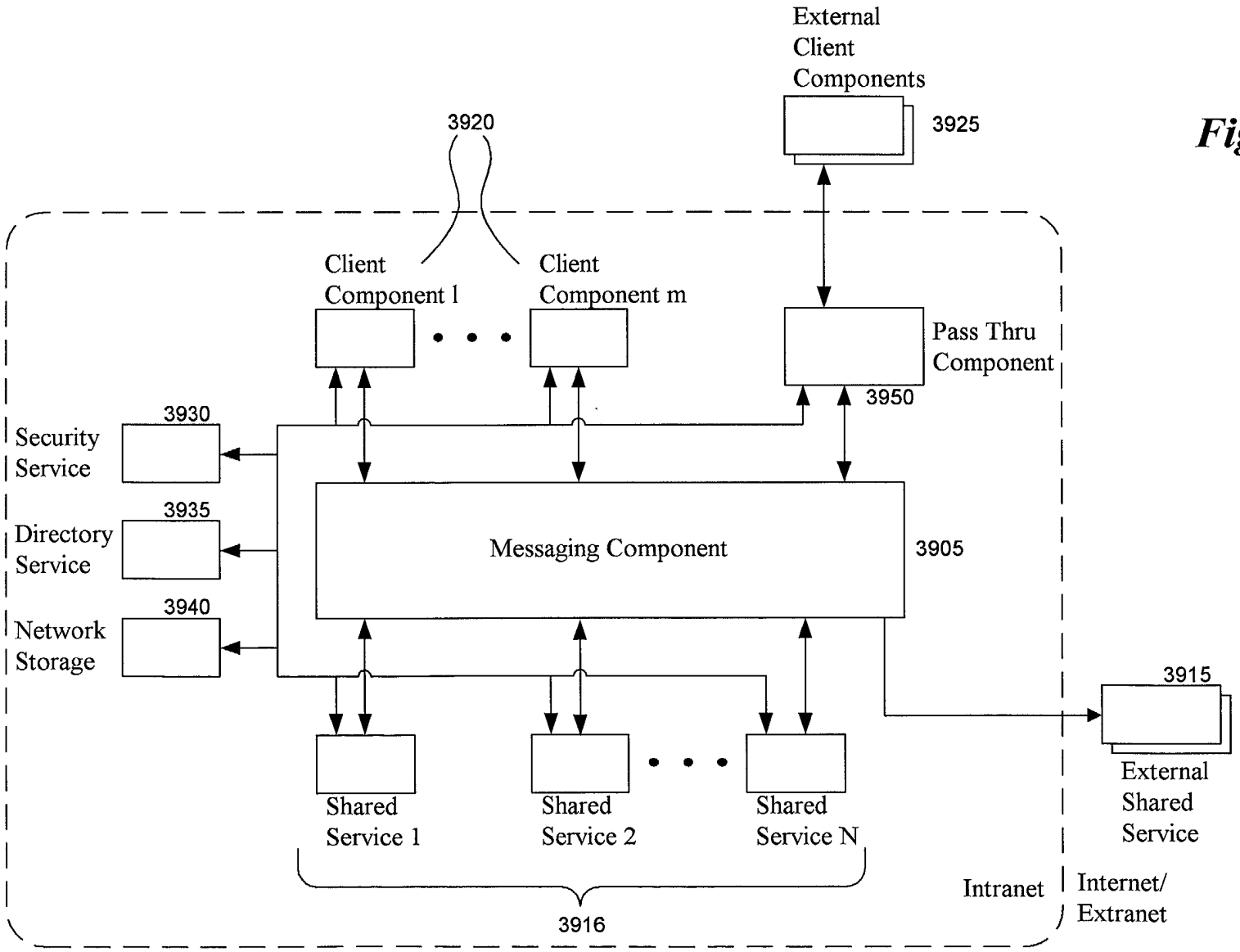

External
Client
Components
3925
3920
Client
Component 1
Client
Component m
Pass Thru
Component
3950
3930
Security
Service
3935
Directory
Service
3940
Network
Storage
Messaging Component
3905
3915
External
Shared
Service
Shared
Service 1
Shared
Service 2
Shared
Service N
Intranet
Internet/
Extranet
3916

Fig. 39

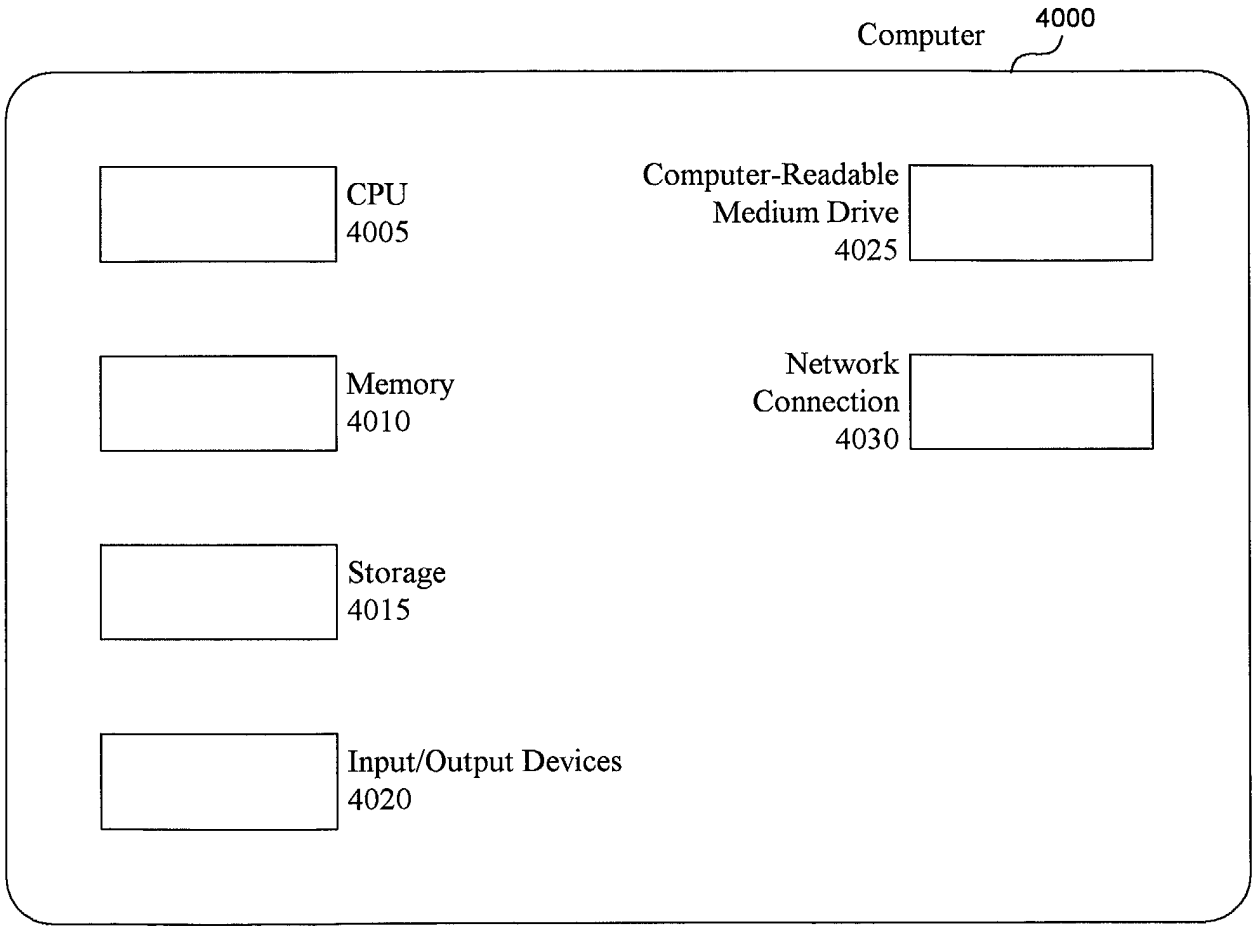

Computer
4000
CPU
4005
Computer-Readable
Medium Drive
4025
Memory
4010
Network
Connection
4030
Storage
4015
Input/Output Devices
4020

Fig. 40

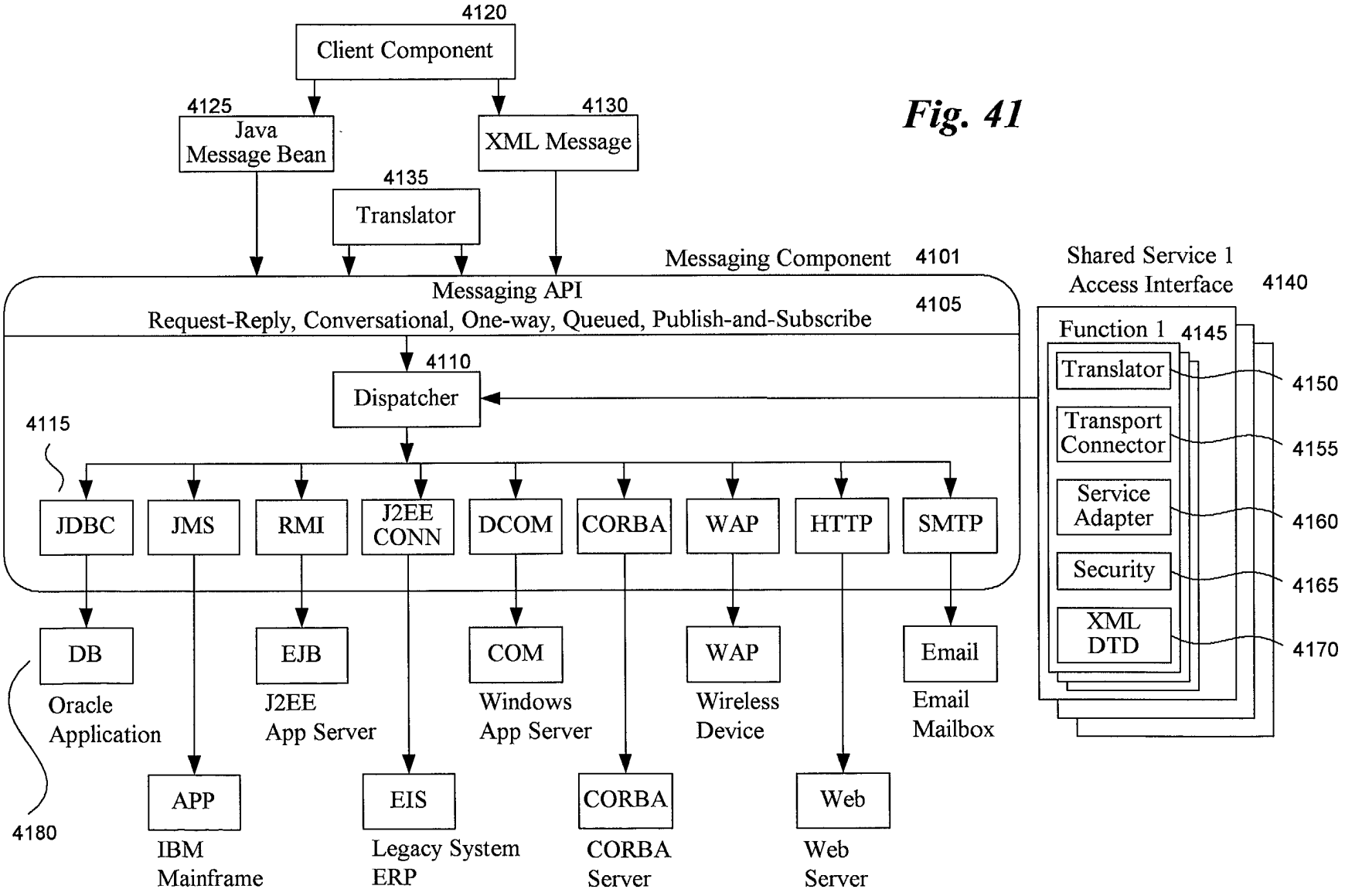

4120
Client Component
4125
Java Message Bean
4130
XML Message
4135
Translator
Messaging Component 4101
Messaging API
Request-Reply, Conversational, One-way, Queued, Publish-and-Subscribe
4105
4110
Dispatcher
4115
JDBC
JMS
RMI
J2EE CONN
DCOM
CORBA
WAP
HTTP
SMTP
DB
Oracle Application
EJB
J2EE App Server
COM
Windows App Server
WAP
Wireless Device
Email
Email Mailbox
4180
APP
IBM Mainframe
EIS
Legacy System ERP
CORBA
CORBA Server
Web
Web Server
Shared Service 1 Access Interface 4140
Function 1 4145
Translator
4150
Transport Connector
4155
Service Adapter
4160
Security
4165
XML DTD
4170

Fig. 41

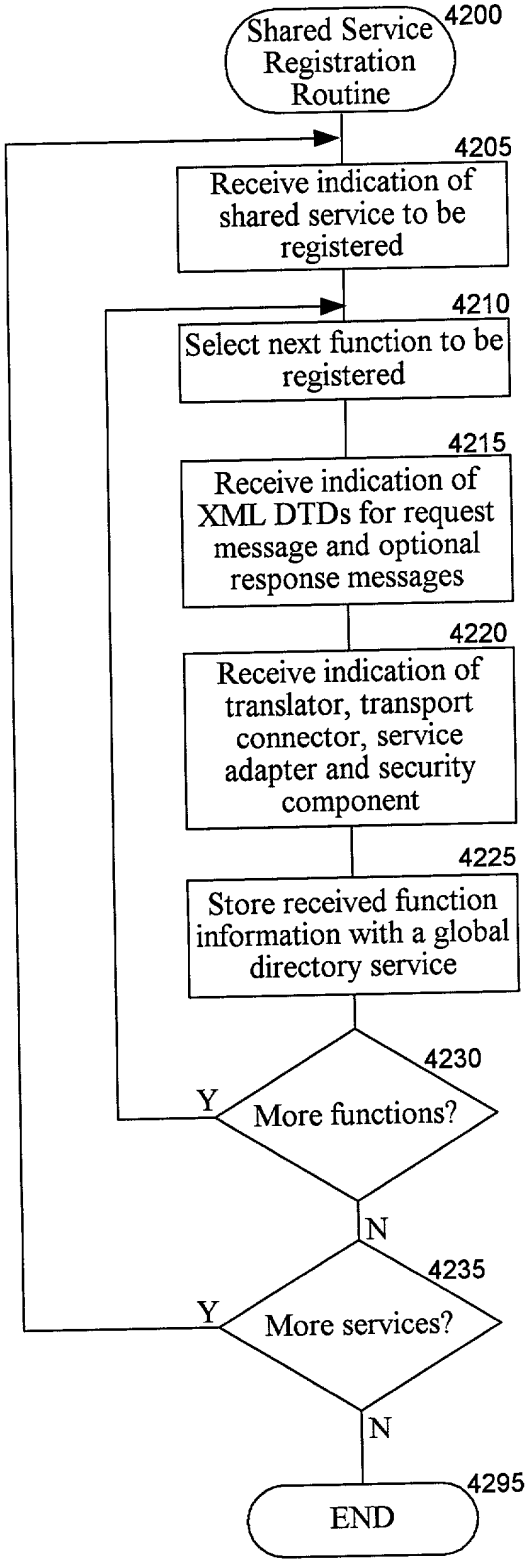

Shared Service Registration Routine
4200
4205
Receive indication of shared service to be registered
4210
Select next function to be registered
4215
Receive indication of XML DTDs for request message and optional response messages
4220
Receive indication of translator, transport connector, service adapter and security component
4225
Store received function information with a global directory service
4230
More functions?
Y
N
4235
More services?
Y
N
END
4295

Fig. 42

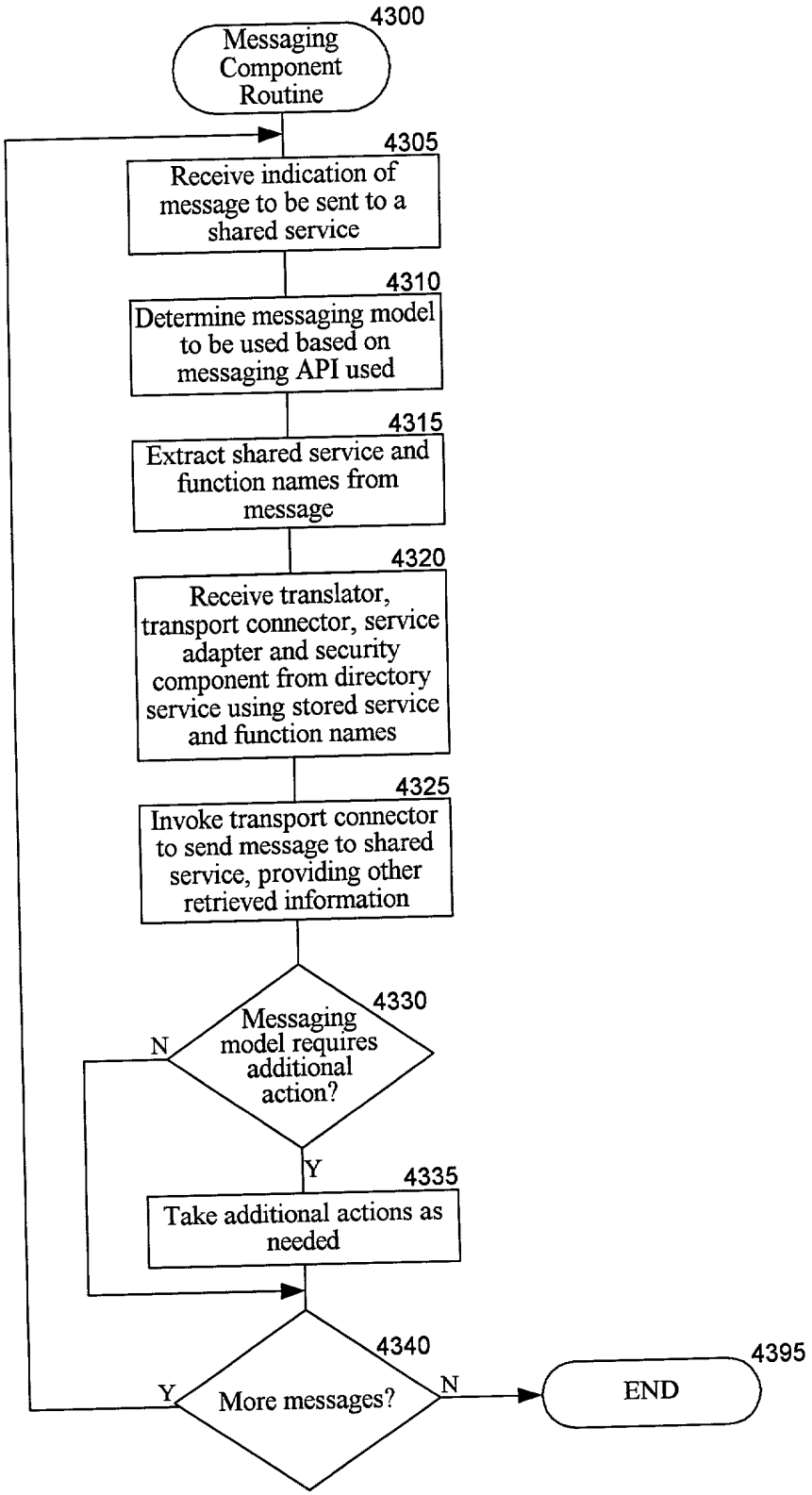

4300
Messaging Component Routine
4305
Receive indication of message to be sent to a shared service
4310
Determine messaging model to be used based on messaging API used
4315
Extract shared service and function names from message
4320
Receive translator, transport connector, service adapter and security component from directory service using stored service and function names
4325
Invoke transport connector to send message to shared service, providing other retrieved information
4330
Messaging model requires additional action?
N
Y
4335
Take additional actions as needed
4340
More messages?
Y
N
4395
END

Fig. 43

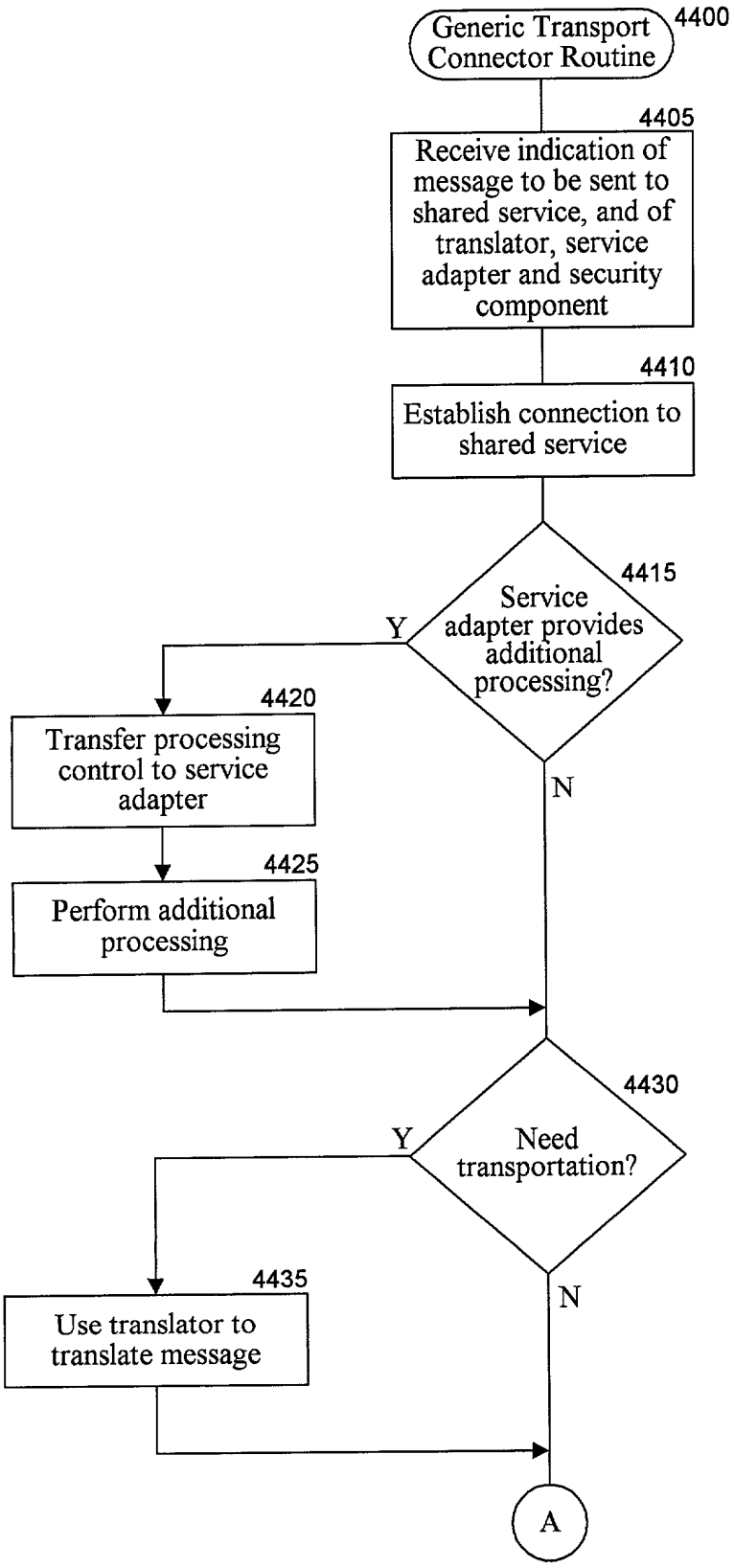

Generic Transport Connector Routine
4400
4405
Receive indication of message to be sent to shared service, and of translator, service adapter and security component
4410
Establish connection to shared service
4415
Service adapter provides additional processing?
Y
N
4420
Transfer processing control to service adapter
4425
Perform additional processing
4430
Need transportation?
Y
N
4435
Use translator to translate message
A

Fig. 44A

Patent Application Publication

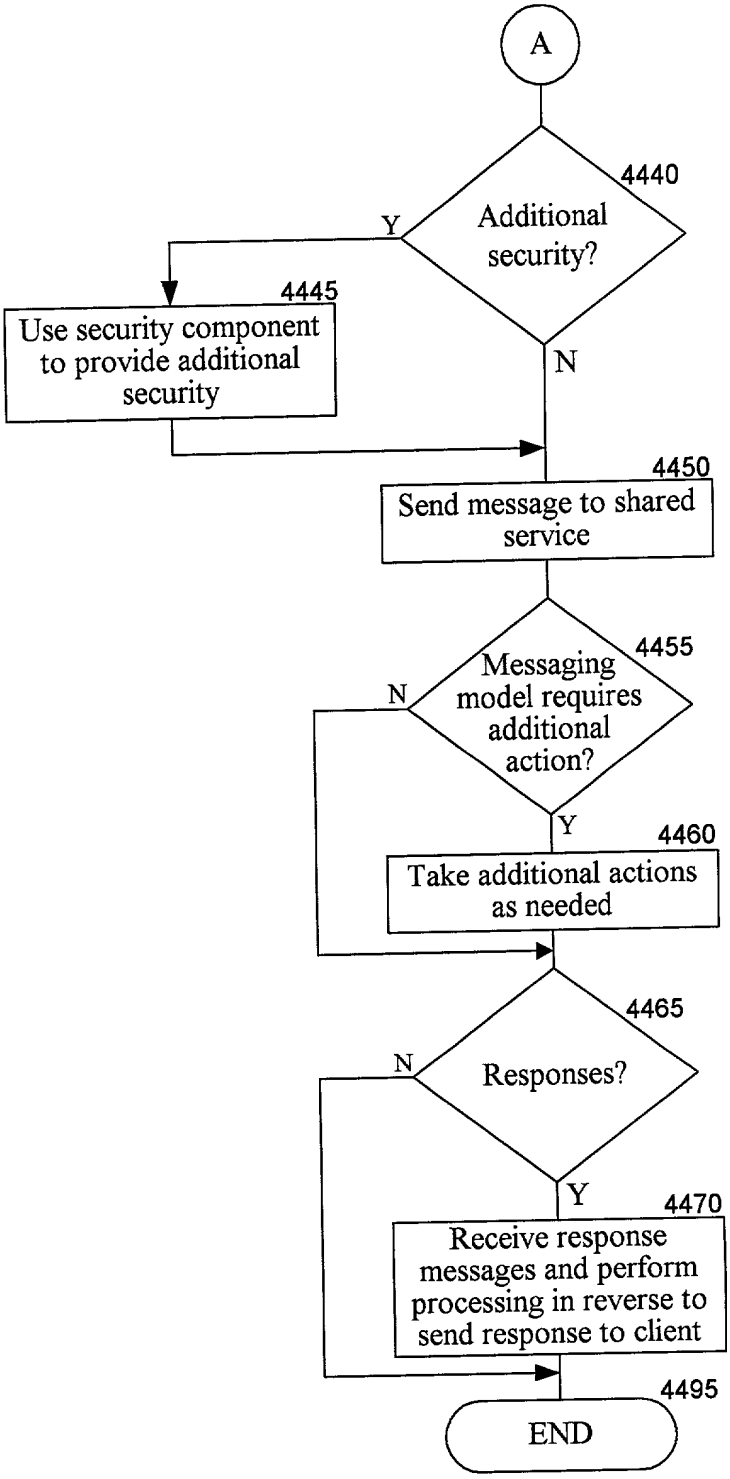


Fig. 44B

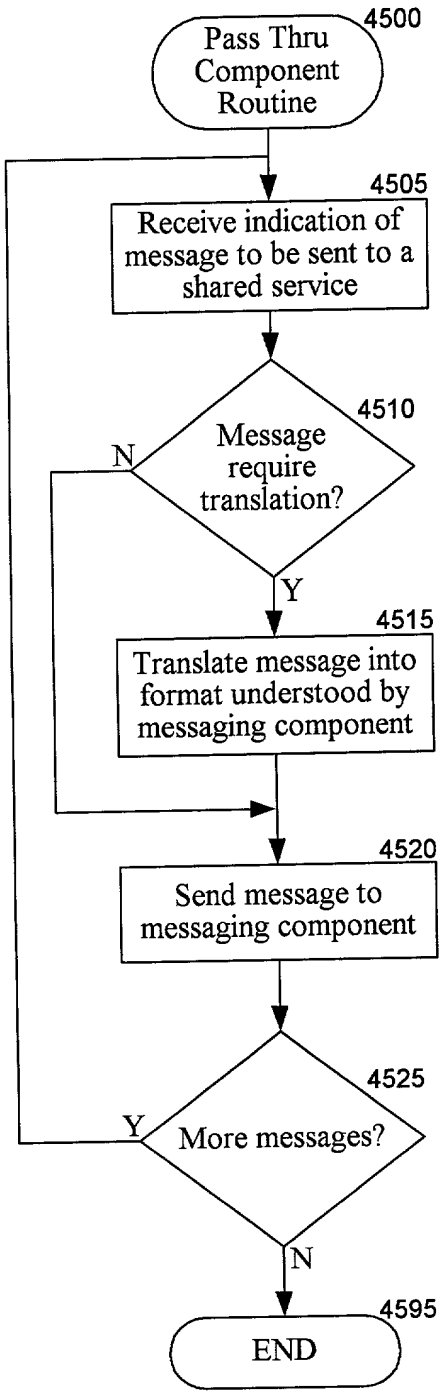

4500
Pass Thru Component Routine
4505
Receive indication of message to be sent to a shared service
4510
Message require translation?
N
Y
4515
Translate message into format understood by messaging component
4520
Send message to messaging component
4525
More messages?
Y
N
4595
END

Fig. 45

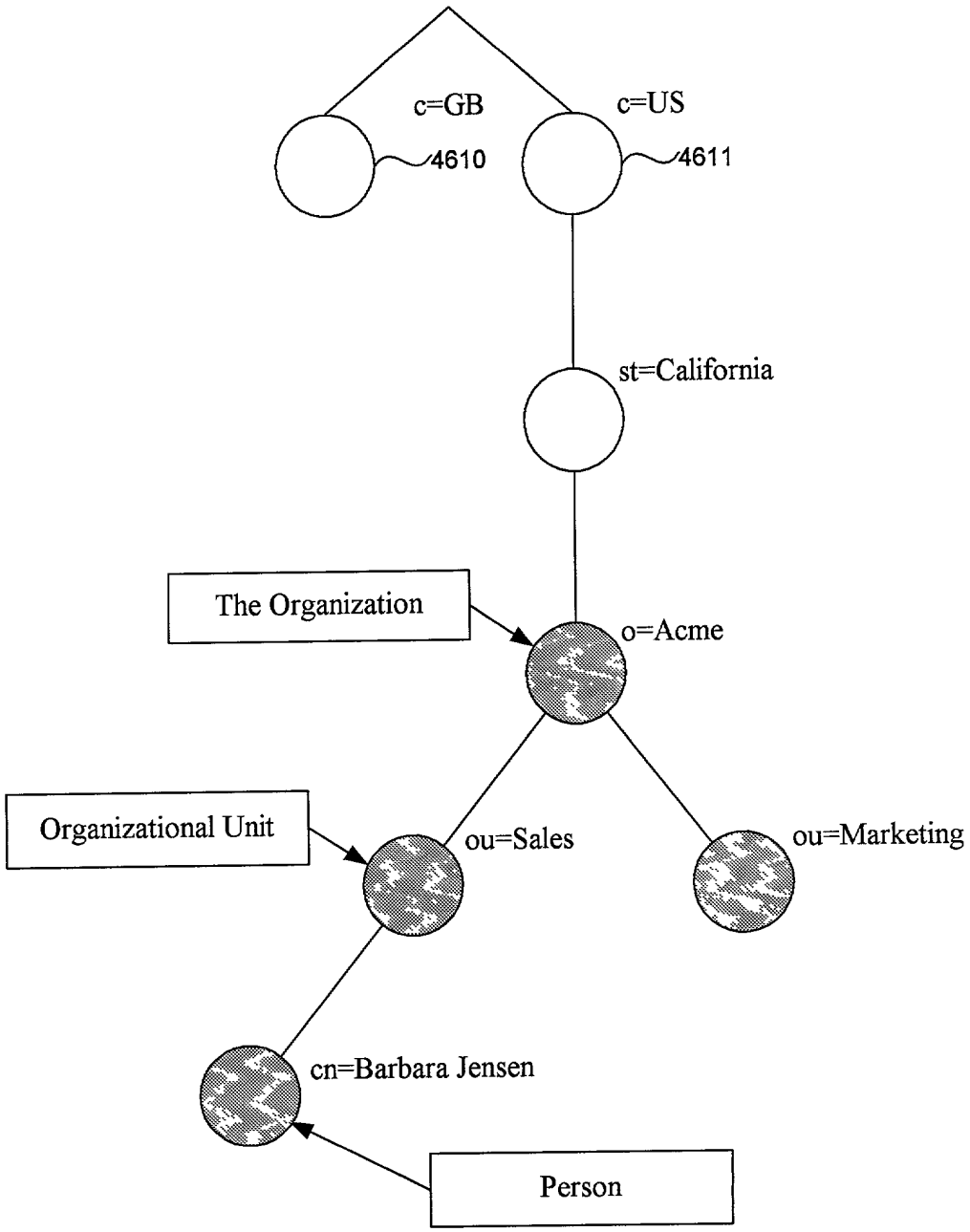

c=GB
c=US
4610
4611
st=California
The Organization
o=Acme
Organizational Unit
ou=Sales
ou=Marketing
cn=Barbara Jensen
Person

Fig. 46A

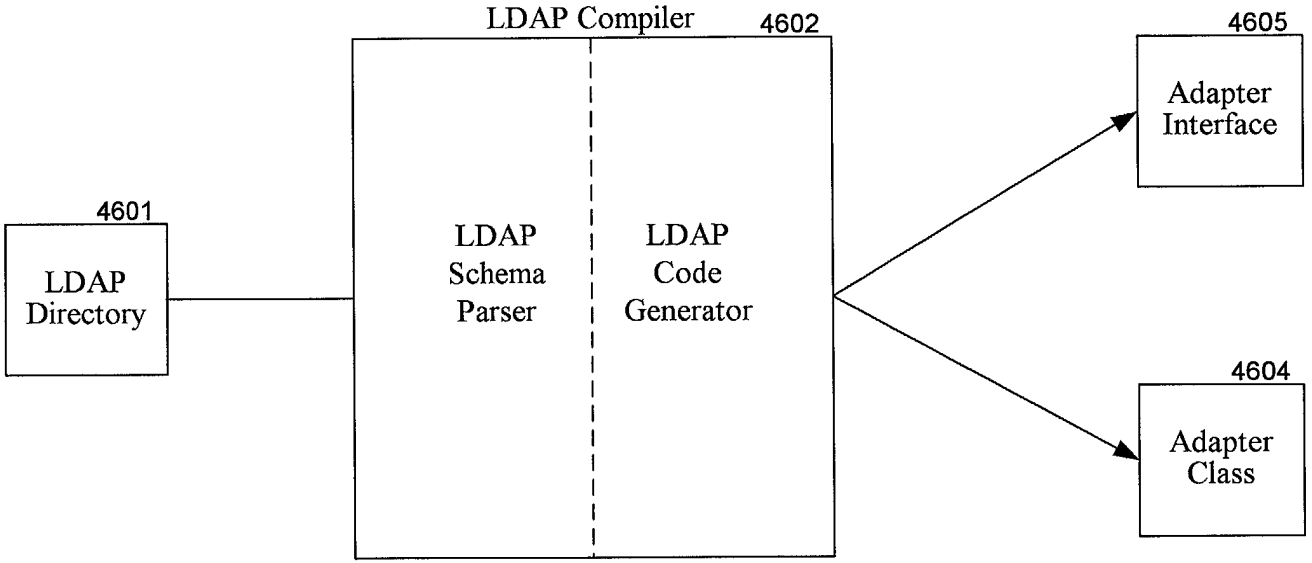

LDAP Compiler
4602
4605
Adapter Interface
4601
LDAP Directory
LDAP Schema Parser
LDAP Code Generator
4604
Adapter Class

Fig. 46B

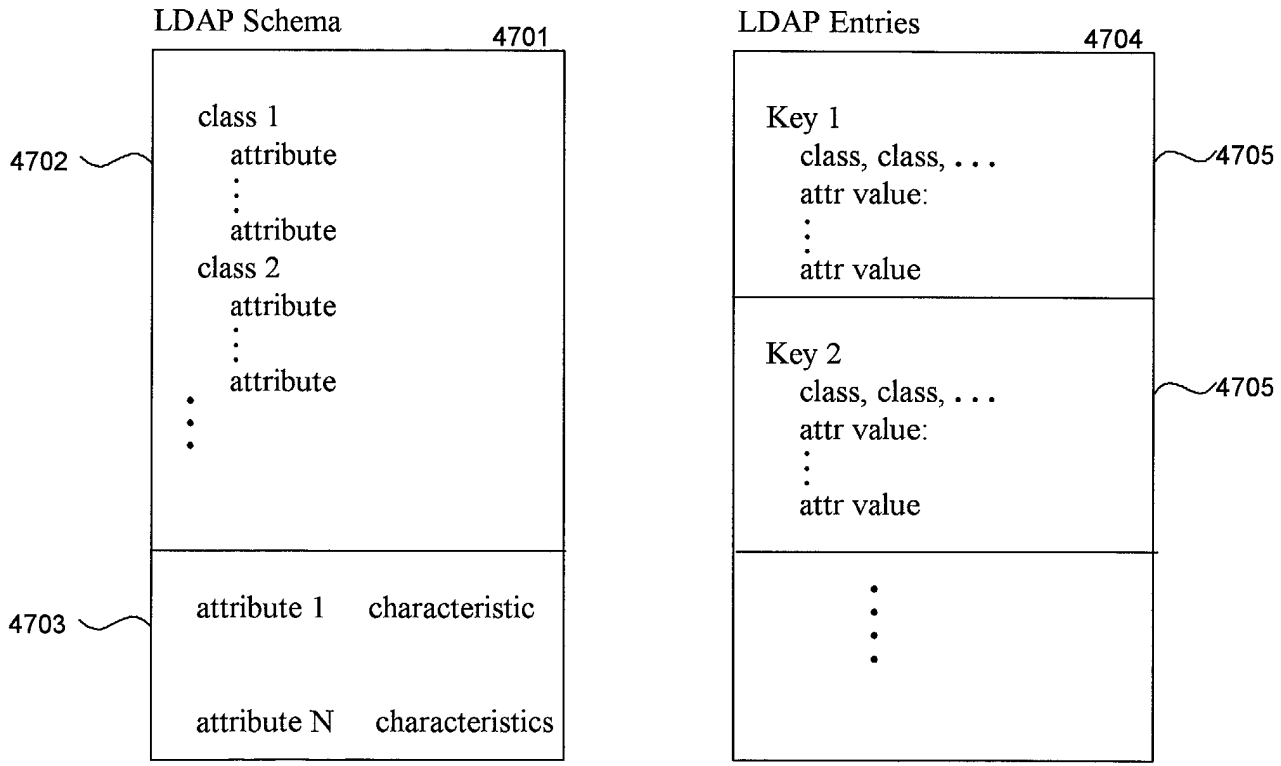

LDAP Schema
4701
class 1
attribute
attribute
class 2
attribute
attribute
4702
4703
attribute 1 characteristic
attribute N characteristics
LDAP Entries
4704
Key 1
class, class, . . .
attr value:
attr value
4705
Key 2
class, class, . . .
attr value:
attr value
4705

Fig. 47

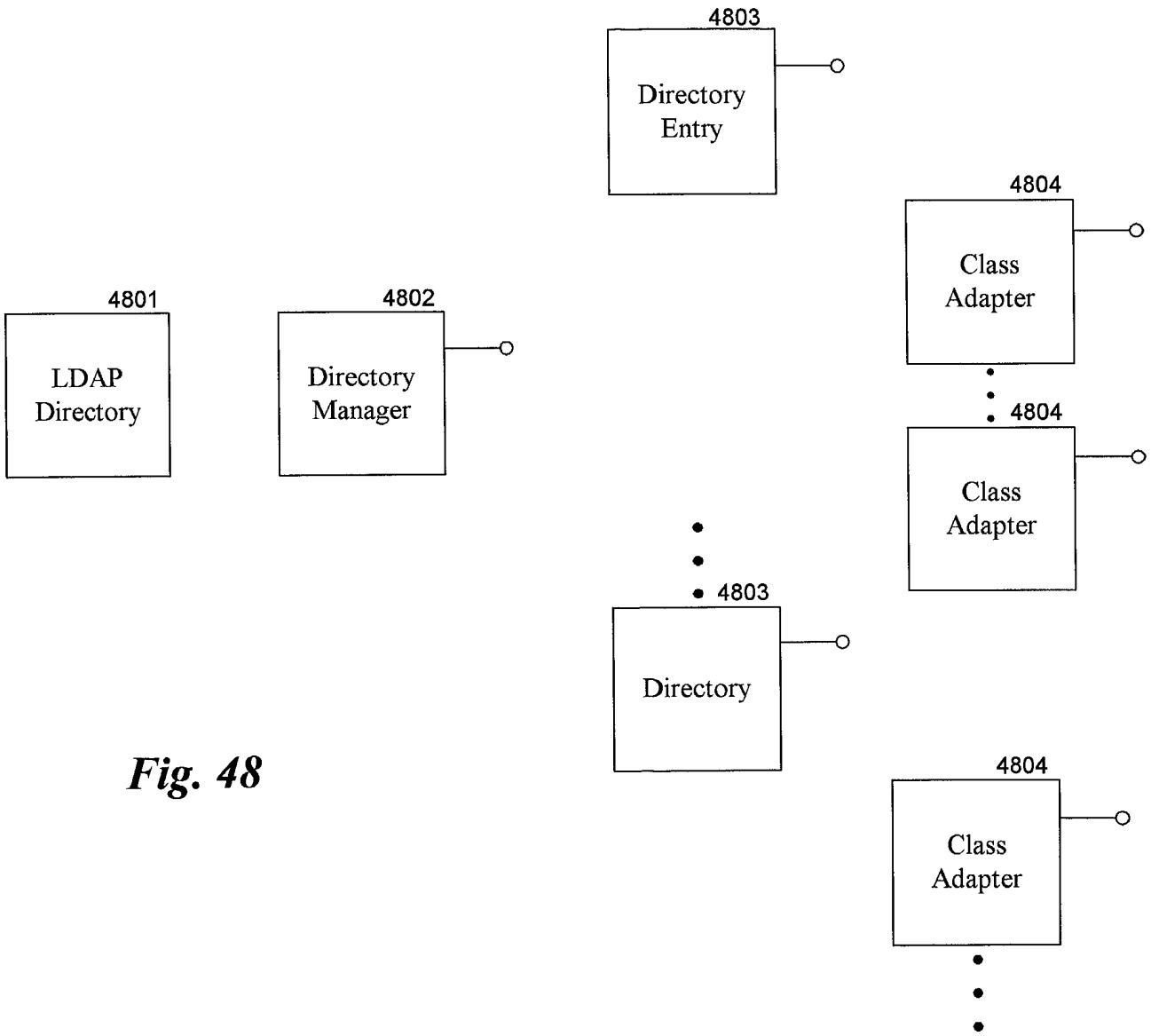

4803
Directory Entry
4804
Class Adapter
4801
LDAP Directory
4802
Directory Manager
4804
Class Adapter
4803
Directory
4804
Class Adapter

Fig. 48

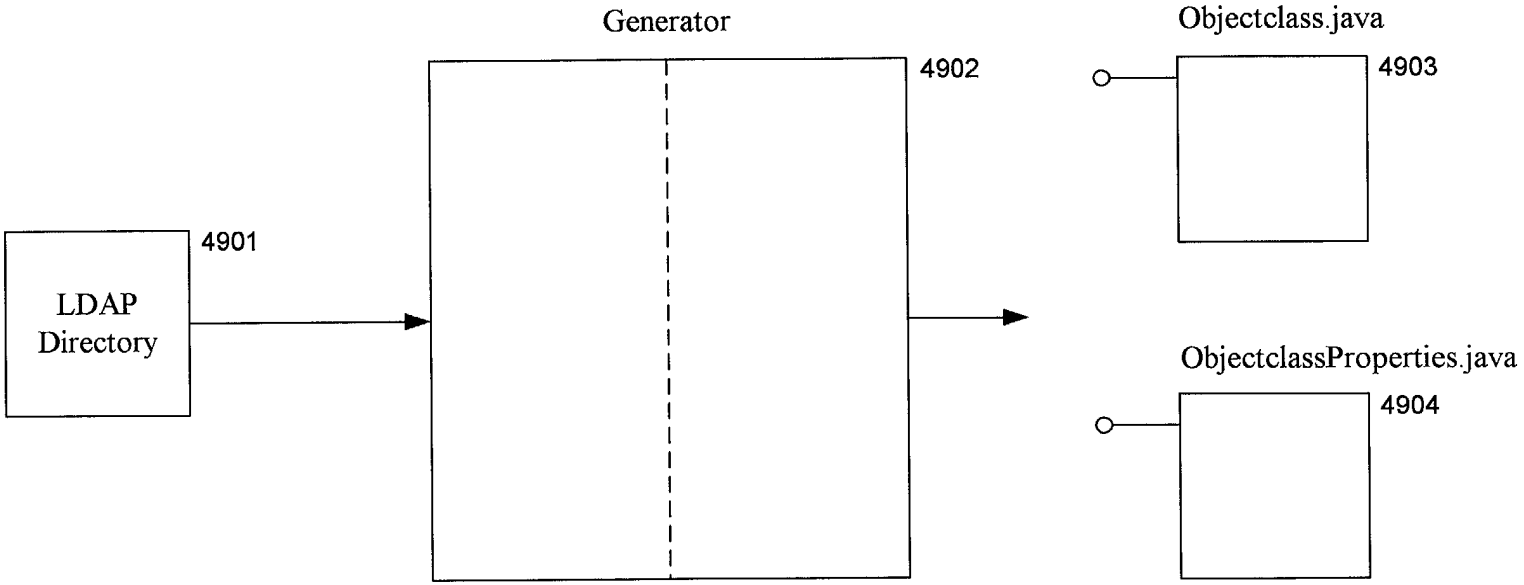

Generator
Objectclass.java
4902
4903
4901
LDAP
Directory
ObjectclassProperties.java
4904

Fig. 49

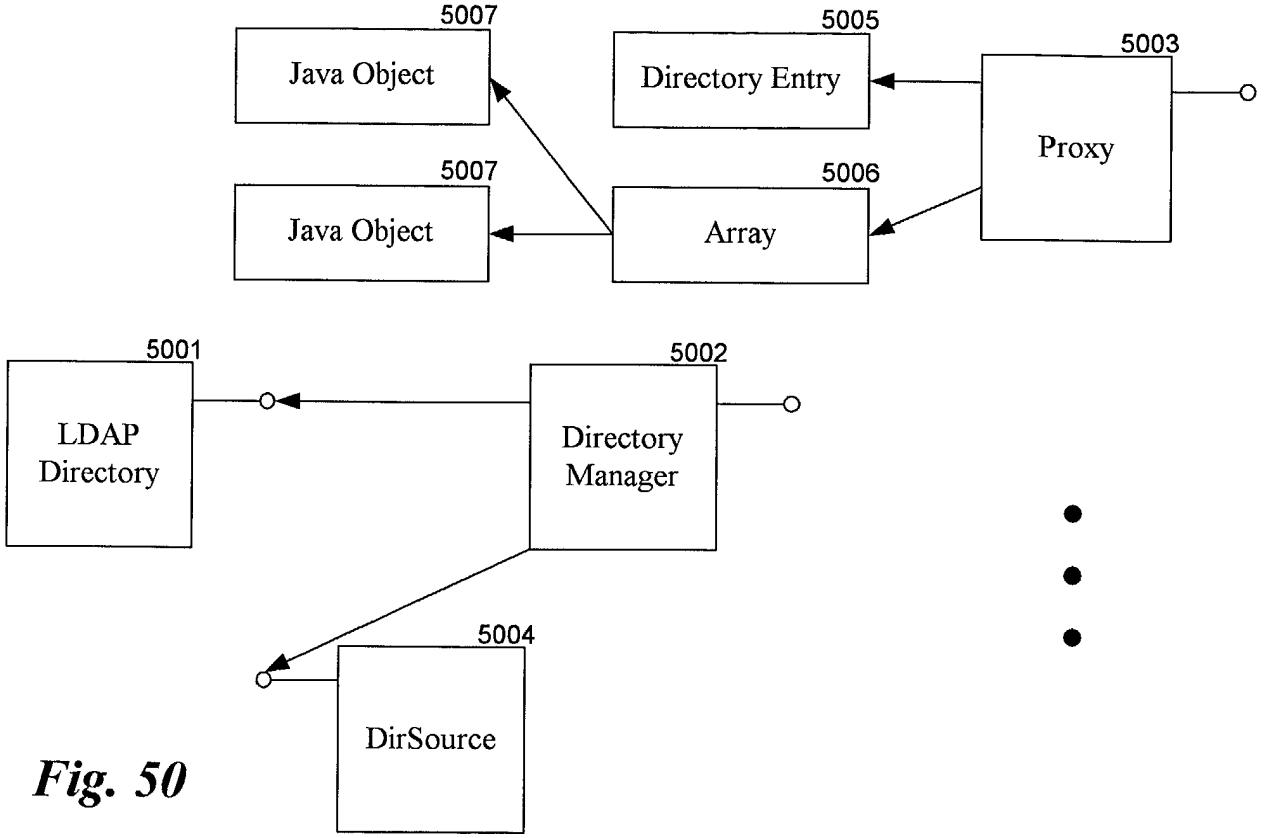
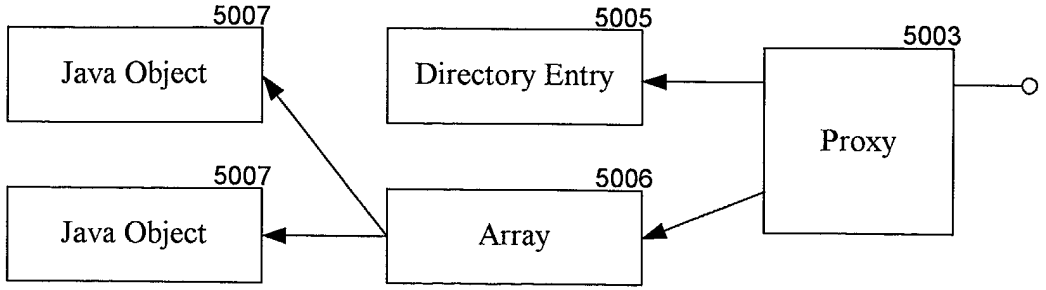

5007
Java Object
5005
Directory Entry
5003
Proxy
5007
Java Object
5006
Array
5001
LDAP Directory
5002
Directory Manager
5004
DirSource

Fig. 50


5007
Java Object
5005
Directory Entry
5003
Proxy
5007
Java Object
5006
Array

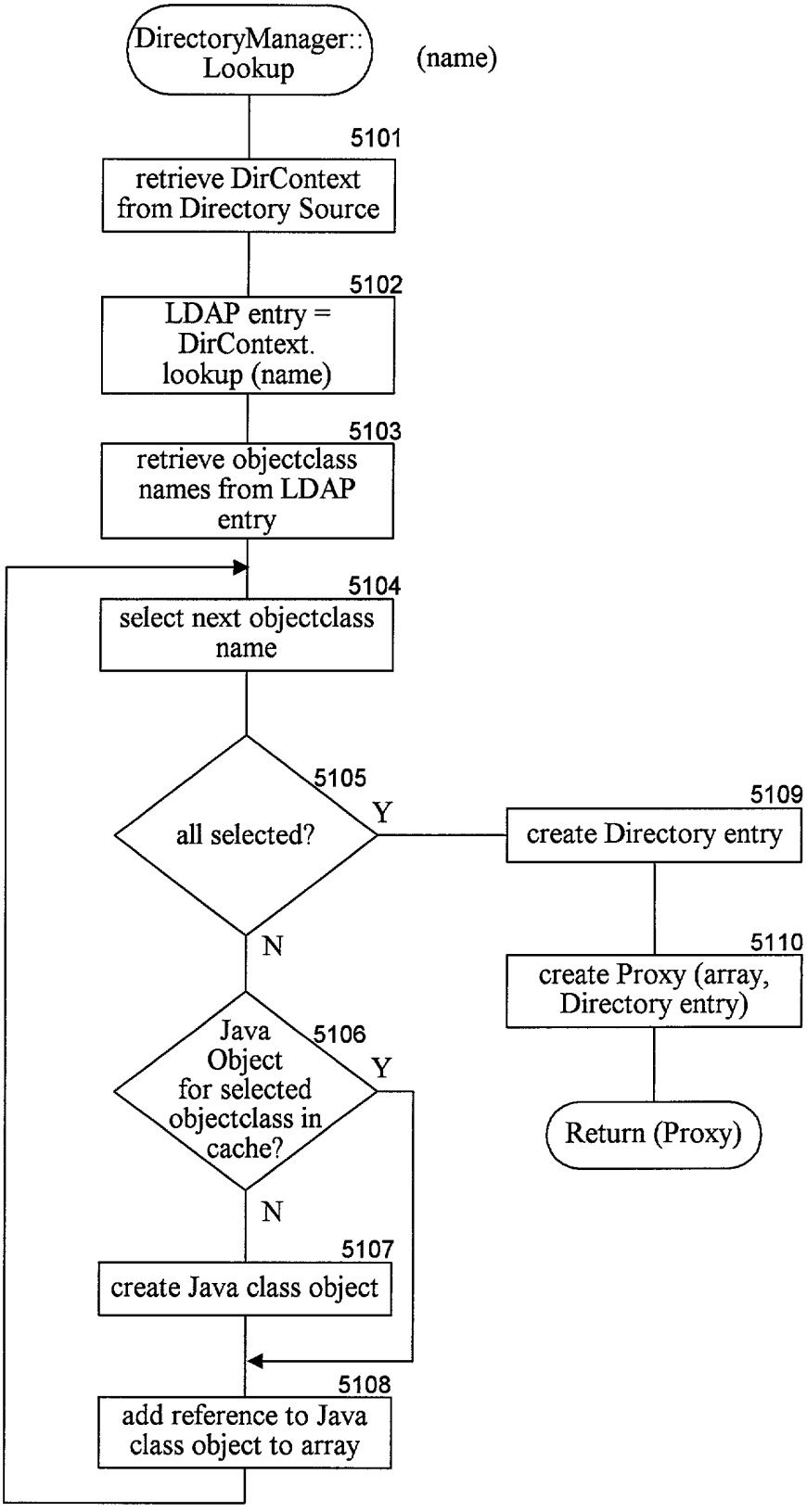

DirectoryManager:: Lookup
(name)
5101
retrieve DirContext from Directory Source
5102
LDAP entry = DirContext. lookup (name)
5103
retrieve objectclass names from LDAP entry
5104
select next objectclass name
5105
all selected?
Y
N
5109
create Directory entry
5110
create Proxy (array, Directory entry)
Return (Proxy)
5106
Java Object for selected objectclass in cache?
Y
N
5107
create Java class object
5108
add reference to Java class object to array

Fig. 51

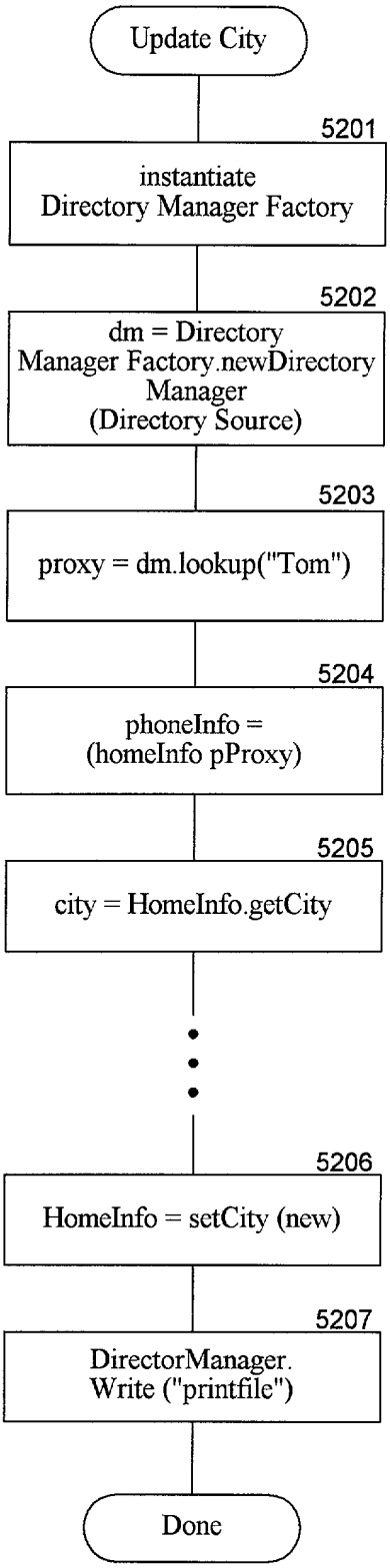

Update City
5201
instantiate
Directory Manager Factory
5202
dm = Directory
Manager Factory.newDirectory
Manager
(Directory Source)
5203
proxy = dm.lookup("Tom")
5204
phoneInfo =
(homeInfo pProxy)
5205
city = HomeInfo.getCity
5206
HomeInfo = setCity (new)
5207
DirectorManager.
Write ("printfile")
Done

Fig. 52

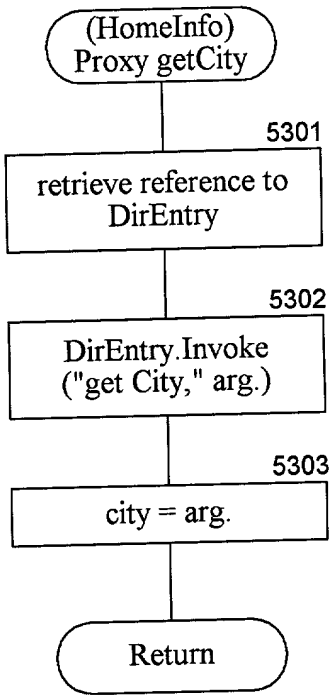
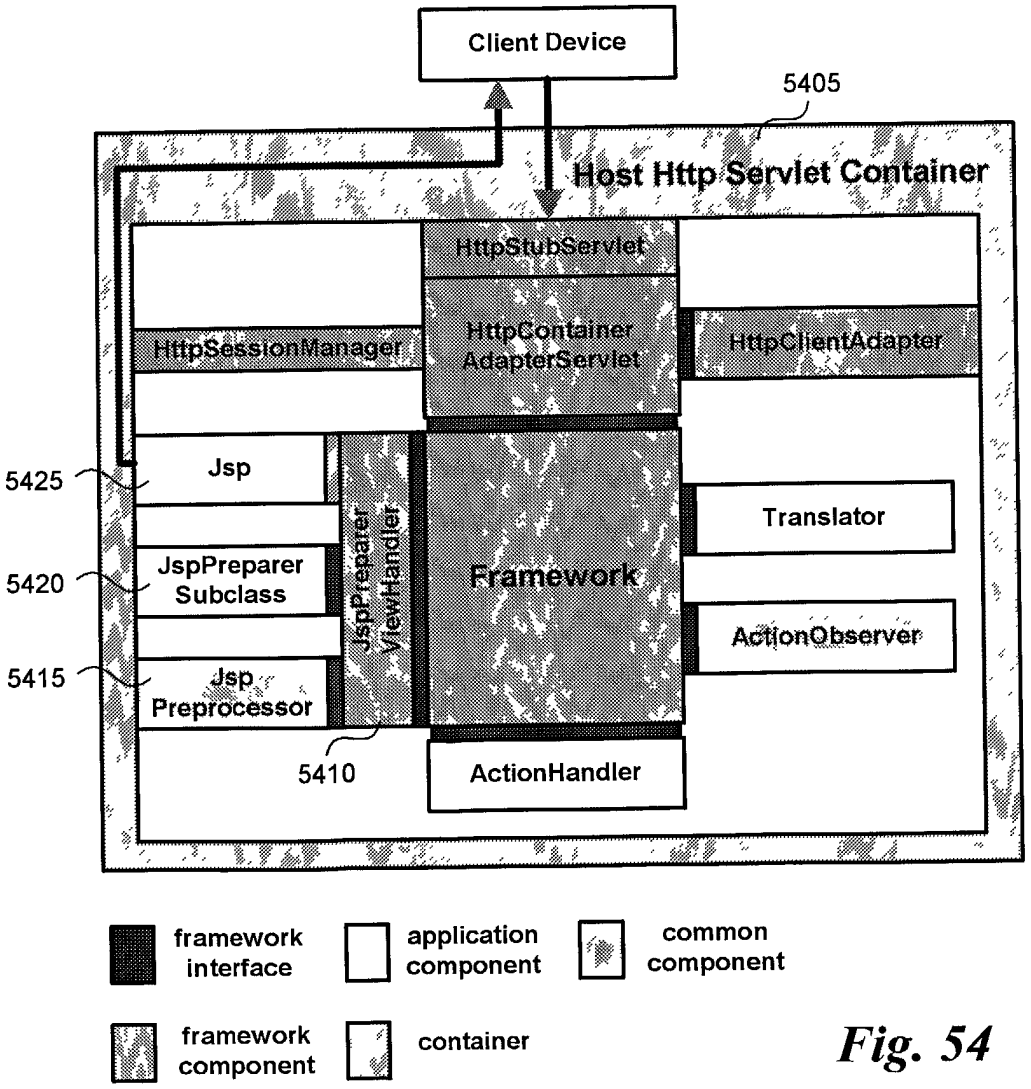

(HomeInfo)
Proxy getCity
5301
retrieve reference to
DirEntry
5302
DirEntry.Invoke
("get City," arg.)
5303
city = arg.
Return

Fig. 53

HTTP Servlet Container



Resource Managers
Bound to the `geps:res/` namespace

- XmlBeanManager
- ConfigManager
- LogManager
- CoreSessionManager
- PrincipalManager
- ObserverManager
- TranslatorManager
- ActionManager
- ViewManager

- JspPreprocessorManager
- PasswordManager
- DirectorySourceManager
- DirectoryManager
- DataSourceManager
- MessagingManager

Fig. 54